SEEK—A Fortran Optimization Program Using a Feasible Directions Gradient Search

M. Savage

National Aeronautics and
Space Administration

U.S. ARMY

RESEARCH LABORATORY

19960202 029

SEEK—A Fortran Optimization Program Using a Feasible Directions Gradient Search

M. Savage
The University of Akron
Akron, Ohio

SEEK - a Fortran Optimization Program

Using a Feasible Directions Gradient Search

M. Savage

The University of Akron

Akron, Ohio 44325

Abstract

This report describes the use of computer program 'SEEK' which works in conjunction with two user-written subroutines and an input data file to perform an optimization procedure on a user's problem. The optimization method uses a modified feasible directions gradient technique. SEEK is written in ANSI standard Fortran 77, has an object size of about 46 K bytes and can be used on a personal computer running DOS. This report describes the use of the program and discusses the optimizing method. The program use is illustrated with four example problems: a bushing design, a helical coil spring design, a gear mesh design and a two-parameter Weibull life-reliability curve fit.

Table of Contents

SUMMARY

This report describes the use of a computer program, 'SEEK' for engineering design optimization. The program is not complete in itself in that it is written to work with problem specific user subroutines and an input data file. It performs a gradient search optimization of the user's problem to find an optimal set of design parameter values. Optimization is performed using a modified feasible directions gradient technique. The program is written in ANSI standard Fortran 77 and has an object size of about 46 K bytes for the optimizing code alone for use on a personal computer running DOS. Its source code is about 1,200 lines in length and its size is 39 K bytes.

In the OPTIMIZATION FORMAT section, the four interface vectors to the procedure are described. These vectors are: 1) the problem constants, 2) the independent design parameters, 3) the constraint bounds, and 4) the objective function terms. The problem constants and independent design parameter values define a specific trial design. In the optimization, the program varies the design parameter values to search for the design which has the best objective function value while satisfying the constraint bounds.

In the PROGRAMMING section, the report describes the two analysis routines, BOUNDS and VALUES which the user must write to evaluate the constrained functions and the objective function. BOUNDS takes as input the constant and design parameter vectors and produces a vector of the constrained function values as its output. VALUES takes as input the same constant and design parameter vectors and produces the objective function value vector as its output. The format and make-up of the input data file which gives the four vectors of constants, design parameters, constraint limits and objective function weighting coefficients are described. In this file, extensive

1

labeling for the four vectors is included. The design verification feature of the optimization program is described also. Once a numerical optimum is found, the program provides the user the opportunity to try alternate designs for comparison purposes.

To illustrate the use of the program, four examples are presented: 1) a bushing design, 2) a helical coil spring design, 3) a spur gear mesh design, and 4) a Weibull data curve fit. The bushing design problem is to find the length and diameter of a bushing to minimize the friction torque in the bearing for a given load and material properties. The spring design is to find the wire diameter and mean coil diameter which support a given alternating load with a required stiffness. Three different design objectives of minimum spring weight, minimum spring height and minimum coil volume are sought. The gear design is to determine the number of pinion teeth, diametral pitch and face width for a compact spur gear set to transmit a given power at a given input speed with a given speed reduction. Two objectives are sought: 1) minimum center distance for a desired life and 2) maximum life for a given center distance. The fourth example shows the fitting of a two-parameter Weibull distribution to life data for a series of identical units tested at the same load.

Each example includes an analysis of the problem, the organization of the optimization variables, the writing of the analysis subroutines and the input data file for a specific problem, the compiling and running of the program, the use of design verification to obtain reasonable values and an interpretation of the output data file. In each case, the design verification opportunity leads to the discovery of a practical solution with near optimal characteristics.

After the examples, the OPTIMIZATION METHOD section presents a description of the gradient search method with its three modes of operation: 1) unconstrained searching, 2) acceptable design searching, and 3) feasible direction searching along one or more design constraints. The unconstrained searching mode employs the gradient in the objective function to improve the design at the fastest rate when possible. The acceptable design searching uses the sum of the gradients in the violated constraints to find the acceptable design region. And, the feasible direction searching mode combines the gradients in the objective function and the violated constraints to improve the objective function value while keeping the trials within the acceptable design region. This section concludes with a description of the program structure and operation.

INTRODUCTION

Optimization is a mathematical process of seeking the most favorable combination of parameters to achieve the best outcome possible [1-3]. In design, one constantly searches for an ideal trade-off of conflicting performance objectives. For aircraft transmissions, for example, we might wish to obtain the lightest transmission which has some minimum acceptable service life, or we may want to maximize the service life at a given transmission weight [4]. These objectives are sought throughout the design and development process with repetitive design descriptions and evaluations - on paper or CAD layout at the design stage and in hardware at the prototype stage. Many of the optimizing, trade-off decisions which develop and improve the product are made by engineers without the help of mathematical models of the product's performance.

Optimization offers the promise of assistance with the difficult trade-off decisions at the early stages of the design, before costly prototypes are constructed and tested [5]. A spectrum of optimization codes have been written to assist in the design of complex structures which can be modeled with large matrices of simultaneous linear equations [2,3]. With an objective computer search through the space of potential designs, we are given a greater chance of determining a truly optimum design.

Unfortunately, many mechanical design problems cannot be described by a set of linear equations. The size of the problem as measured by the number of independent design parameters and the constraints which are to be satisfied may be small. But the complexity of the problem may be large due to the non-linear and often discontinuous nature of the design property and constraint relations. This combination keeps the optimization of mechanical designs in

4

the realm of an art which requires considerable engineering insight to complement the available mathematical models and computer solutions [6].

In modeling mechanical systems, one is often confronted with the choice of obtaining exact solutions to an approximate model or obtaining approximate solutions to an exact model. With the designer active in the process, rapid solutions of an approximate model can lead to practical optimal designs even when the mathematical optimum contains some flaws [7]. Modified gradient optimization techniques such as the feasible directions method are quite powerful and rapid when they have continuous models in which gradients can be calculated. The method of this optimization is to have the engineer write Fortran subroutines which model the design with continuous properties as functions of the constants and independent design parameters which define the design. The optimizer can then find the optimal solution to this 'ideal' problem and report it to the engineer to allow a check of alternate designs which satisfy additional constraints of practicality. The end result is a practical, optimal design.

This report describes the use and background of a Fortran program, SEEK, which is written to assist the mechanical designer in developing balanced optimal designs. The intent of the program is to keep the engineer in the process while providing a systematic search of potential designs. In doing this, it allows the engineer to use the mathematical models of the optimization to evaluate near optimal, practical designs. SEEK, which requires two user-written analysis subroutines, reads its input from an ASCII data file and writes the output of the optimization both to the screen and to an output ASCII log file. To document the optimization clearly, the input

5

data file includes a significant amount of text to describe the numerical values in the output file.

The report includes an OPTIMIZATION FORMAT section which describes the basic format of the optimization problem: the constants, design parameters, constraint bounds and objective function; as well as a PROGRAMMING section, which describes the programming required to prepare SEEK for use in an interactive design session. This section describes: the two analysis subroutines, the input data file, the use of the program with changes to the input file and design verification in the interactive session.

To demonstrate the power and ease of use of this optimization procedure, several small design examples follow in the next sections: a bushing, a spring, a spur gear mesh, and the fitting of a two parameter Weibull distribution to experimental test data.

An OPTIMIZATION METHOD section follows which describes the structure and operation of the optimization code. This code is small with 1200 lines and less than 40 k bytes so the optimization can be performed on a personal computer running with DOS. The speed of a 486 machine may become attractive for the more complex analysis models.

OPTIMIZATION FORMAT

An optimization problem may be formulated as a constrained search in terms of four vectors and two sets of relations. In this formulation, only inequality relations are used for the constraints. The four vectors are: 1) the constants of the problem which do not change for a given design, 2) the parameters which define a design and which are the sought values, 3) the constraint values which may be upper or lower bounds on properties of the design, and 4) the objective function's weighting coefficients.

In this formulation, at least two Fortran subroutines are needed: 1) BOUNDS which evaluates the constrained variable values in terms of the constants and design parameter values and 2) VALUES which evaluates the objective function's value for a given set of constants and design parameter values. These two subroutines combine with the input data file to define the specific problem for optimization. The gradient calculations which perform the optimization by calling these two subroutines repeatedly and the input and output file interfacing are contained in the SEEK Fortran code.

Constants

Each problem is defined by a series of constant values such as: size, power level, speed of operation, elastic modulus, material strength or requested service life. These constants are fixed for all trial designs of the optimization effort, and the constrained properties and objective function values are direct functions of them. The constants may change for a different design using the same analysis subroutines, however. For example, designs made of steel will have different properties from those made of nylon, yet each steel design will have the same material stiffness and strength as the other steel designs.

7

The program will read these constants and their labels from the ASCII input file, store them in arrays and use the values whenever the constrained property or objective function values are calculated.

Parameters

In each problem, we are searching for a set of parameter values which optimize the objective function to either a minimum or a maximum value. These parameters are the second vector entered in the input data file. The optimization scheme proceeds by analyzing repeated trials until it selects one for which the analysis yields an optimum objective function value. So the values entered for the design parameters include an initial value for each parameter for the first trial analysis. This vector also includes upper and lower values for each design parameter. These upper and lower values serve to establish the relative sensitivity of the parameter for the gradient searches, but do not limit the value itself. By increasing the span between the upper and lower values for a design parameter, the user can increase the sensitivity of that parameter in the design search. If it appears that a design parameter is not changing as the optimizer seeks out better designs, increasing this span between the upper and lower values will increase its tendency to change in future optimization runs.

After reading these parameter values and their labels from the input file, the program will store them in arrays, use the parameter ranges to set relative sensitivities which will not change throughout the optimization and place the initial parameter values into the parameter array. The parameter array will change throughout the operation of the program until it contains the values of the parameters which optimize the objective function.

8

Constraints

Limiting each design is a series of constraints on the properties of the design. These constraints may be applied directly to one or more of the parameters such as the thickness of a beam or they may be applied to a calculated property of the design such as the maximum stress in the beam. The constraints of this algorithm are inequality constraints. In the general optimization formulation, two types of constraints are possible: inequality and equality. Gradient search algorithms require a continuum of parameter and property values in which to move around in search of the optimum. Inequality constraints provide boundaries to the design space but do not diminish it.

Equality constraints reduce the space by one dimension. Each equality constraint transforms one independent design parameter into a dependent design parameter. There are two ways to include an equality constraint in this algorithm: 1) reduce the design space by one, or 2) enter the equality constraint into the data as an inequality constraint.

The first method is the best because it simplifies the calculations making the optimization more direct and faster. If the width of a rectangular spring is always seven times its height, then one can remove the width from the list of independent design parameters and set it equal to seven times the height in the calculations. This reduces all gradient calculations by one element and leaves a full design space for the remaining parameters' gradient searches.

The second method may be easier to implement if the equality constraint is not tied to the parameters directly. By entering it as an inequality constraint, one leaves the design space at its larger dimensional size and cuts it in half with the bounding value. Since the unbounded optimal design

9

would probably lie off this constraint, placing this bound between the unbounded optimum and the acceptable design space will place the bounded optimal design right on this constraint and thus actually satisfy the equality constraint.

The program will read these constraints, their directions and their labels from the input file, place them in arrays and use them in the subroutine BOUNDS, which is provided by the user, to limit the acceptable design space throughout the optimization.

Objective Function

In each optimization, some property or combination of properties, called the objective function, is to be minimized or maximized. The weighting coefficients of these properties are the last vector entered in the input data file. These coefficients may be percentages, unit conversions to place the properties in the same dimensions or they may be switches such as zero and one to change the optimization in the data file by changing the sought objective. The assumption is that the objective function to be optimized may be expressed as a linear sum of terms, each with its own weighting coefficient. The weighting coefficients and direction of optimization will remain fixed throughout the optimization.

The program will read these coefficients and their labels from the end of the input data file, place them in arrays and use the coefficients to modify the objective function property values. These values are calculated in subroutine VALUES which is provided by the user.

PROGRAMMING

With these four vectors defined and labeled, the program starts the output file with an echo of the input data to document the optimization problem. It then proceeds to seek an optimal design with the modified feasible directions gradient method. Using gradients in the objective function and in the violated constraints, the program can move from an initial design which does not satisfy the design constraints to designs which are valid. It can also improve a valid initial design to obtain an optimum design within the assumptions of the model.

Once an optimum design is found, the program prints: the found design parameters, the objective function value with its component function values, and the constraints with both their design and limit values. The program then offers the user the opportunity to try alternate designs. On receiving the revised design parameter values, the program uses subroutines BOUNDS and VALUES to check this design, prints out its properties and offers the user the chance to try another alternate design. All design trials are printed to the screen and the ASCII output log file.

Analysis Subroutines

To model a problem, the program needs two analysis subroutines: BOUNDS and VALUES. These subroutines are problem specific and should match the input data. Subroutine BOUNDS calculates the constrained property values for each design trial and gradient perturbation as direct functions of the constants and design parameters only. Data are passed to BOUNDS with two dynamically dimensioned arrays: CONST(NCO) for the constants and X(NX) for the design parameters, and the constraint value results are returned to the program in the array VCSTR(NCS). Subroutine VALUES calculates the objective function

11

values also as direct functions of the constants and design parameters only. Data are passed to VALUES with the same two arrays: CONST(NCO) for the constants and X(NX) for the design parameters, and the object function values are returned in the array OBJECT(NOB).

Both subroutines must be able to determine their outputs as continuous functions over the range of design parameters used. Since small perturbations are given to the design parameters to determine corresponding changes in the objective function value and in the constrained variable values throughout the design search, the subroutine calculations must be defined and continuous. Discrete parameter requirements such as integer tooth numbers for gears and standard component sizes can be added by the user in the verification stage of the optimization process. They cannot be included in the simulation model itself.

These subroutines may contain formulas, interpolated data, iterations or other subroutines as long as the resulting calculations yield continuous functions of the design parameters. If the subroutines call other subroutines, they should not have the same names as those subroutines included in the optimizing part of SEEK, which are listed in Table 1.

Common blocks may be used by the subroutines, but SEEK uses four common blocks, which should not be altered: CURVE, PAR, VAR, and UNITS. One of these common blocks, UNITS, contains four integer variables, NW, NR, NF and ND. These are the file numbers for writing and reading to the interface devices. NW identifies the screen, NR identifies the keyboard, NF identifies the output file and ND identifies the input file. The user may add additional information to the output with these file numbers, bearing in mind that BOUNDS and VALUES are called many times by SEEK in the optimization process.

12

Table 1

Subroutines of Program SEEK

Line	Name	Function
756	BACK	Search for Acceptable Design Space
983	BOUNCE	Find Gradient Sum of the Violated Constraints
1207	BOUNDS	User Supplied Constraint Analysis
1113	CHECK	Test for Constraint Violation
1033	GRADNT	Evaluate a Gradient
1174	MERIT	Evaluate the Objective Function Sum
724	RESIZE	Unscale the Design Parameters into Real Units
912	SCAN	Increment the Design in the Acceptable Design Space
874	SCOUT	Try a New Design Position And Check the Constraints
740	SIZE	Scale the Design Parameters to Unit Space
1088	UNIT	Normalize a Vector
1210+	VALUES	User Supplied Objective Function Analysis
1146	WALL	Evaluate a Specified Constraint

Input Data File

Coordinated with these two required subroutines is the ASCII input data file, which is described in Appendix A - SEEK.DOC. The initial line in the data file is a text line of fifty characters or less which describes the design being optimized. This is followed by a line which contains a single number, NCO, - the number of constant values to follow, which is the first vector in the data file. Each optimization constant is then entered in a set of three lines: 1) the numerical value, 2) the name of the constant in thirty characters or less, and 3) the units for the constant in twelve characters or less. With this information, the program will label the constant values whenever it prints them.

Following the constants in the input data file is the list of independent design parameters, which is the second vector. After the last constant has been listed, the next line is once again a single number, NX, - the number of parameter values to follow. Each parameter is then entered in a set of three lines: 1) three numerical values - a low estimate for the parameter, a high estimate and an initial estimate; 2) the name of the parameter in thirty characters or less; and 3) the units for the parameter in twelve characters or less.

The list of constraint bounds is the third vector. After the last parameter has been listed, the next line is a single number, NCS, - the number of constraint bounds to follow. Each bound is then entered in a set of three lines: 1) the word 'UPPER' or 'LOWER' followed by the numerical value including its decimal point, 2) the name of the constraint in thirty characters or less, and 3) the units of the constraint in twelve characters or less.

14

Finally, the list of weighting coefficients is the fourth vector entered. After the last constraint has been listed, the next line is a three letter prefix, 'MIN' or 'MAX', which describes the direction of optimization. This is followed by a line with a single number, NOB, - the number of weighting coefficients to follow. Each coefficient is then entered in a set of three lines: 1) the numerical value, 2) the name of the property in thirty characters or less, and 3) the units for the property in twelve characters or less.

Design Verification

As described at the start of this section, the use of SEEK is somewhat interactive. Because the user must add at least two subroutines to the program in addition to the input data file, the combined program must be compiled separately for each optimization application. After writing the analysis subroutines with the proper dynamic dimensioning in the calling arguments, the user may add subroutines to the end of the source code for SEEK and compile the program in the environment in which it is to be run. The compiler should be a Fortran 77 compiler of which there are several PC versions available. Once compiled and linked into an executable program, the optimizer can be run with the matching data file to find an optimal design.

With the interaction of the data file and the analysis subroutines, the user may change the way an optimization is conducted through small changes in the data file. By using one and zero as weighting coefficients to an objective function that contains totally different terms and by switching constraints from UPPER to LOWER or changing their values to make them active or inactive, one can change an optimization significantly.

For example, one could have a transmission life optimization program which included bounds on the transmission size and life as well as terms in the objective function for size and life [8]. By requesting that the size be less than some value, that the life be greater than zero and by having a weighting coefficient of one for life and zero for size and by selecting 'MAX' in the input data file, one would have an optimization that would maximize the life of the transmission within a given acceptable size. Shifting the requests in the input file to request that the size be greater than zero, that the life be greater than some desired value, and by having a weighting coefficient of zero for life and one for size with 'MIN' selected in the input data file would minimize the size of the transmission for the requested service life.

Smaller changes in limit values or problem constants could change the size of a requested design or some other feature without requiring a change in the compiled program. As stated earlier, the program generates a complete log file of the obtained designs and the verified designs in response to keyboard input after an 'ideal' design has been found and written to the screen and the log file. Speed of execution of the program is entirely dependent upon the complexity of the analysis models. Small optimization programs can run quickly on the personal computer. The following four sections will demonstrate the use of SEEK for several different design optimizations.

16

BUSHING OPTIMIZATION

Four examples of increasing complexity will be presented to illustrate the capability of SEEK. The first example is that of the design of a low-speed bearing to support a radial shaft load. For this application, the simplest bearing is a bushing which is defined by its material, length and diameter.

Theory

Consider the design of a bushing to support a radial shaft load. With little or no lubrication, a bushing's capacity is both strength and power limited [9]. By constraining the bearing length to be less than or equal to some percentage of the shaft diameter, one can treat the radial load as supported uniformly over the length of the bushing. Thus:

$$\frac{L}{D} < \beta \tag{1}$$

The nominal contact pressure in the bushing can then be taken as:

$$P = \frac{F}{L\,D} \tag{2}$$

where the pressure, P, is measured in MPa; the load, F, is in Newtons; and the length, L, and diameter, D, are in mm. And the sliding velocity in the bushing, V_s, measured in m/s is:

$$V_s = \frac{D}{2}\,\Omega\left(\frac{2\,\pi}{60}\right)10^{-3} \tag{3}$$

where the shaft speed, Ω, is in RPM. The strength limit on the bushing is thus:

17

$$P \leq P_{max} \qquad (4)$$

The power limit on the bushing, which is proportional by the coefficient of friction to the power lost in the bushing, is the PV factor of:

$$P V_s = \left(\frac{F}{L\,D} \right) \left(\frac{D}{2} \; \Omega \left(\frac{2\,\pi}{60} \right) \right) 10^{-3} \qquad (5)$$

$$P V_s = \frac{F\,\Omega}{L} \left(\frac{\pi}{60} \right) 10^{-3} \leq PV_{max} \qquad (6)$$

Figure 1 is a graph of the contact pressure in a bushing versus the contact sliding velocity which shows the regions of the two pressure limits. Acceptable designs have pressures lower than the plotted values. As the speed increases, the power limit becomes active and restricts the design to lower and lower acceptable pressures. Values for these limits for both metallic and non-metallic materials are readily available [10,11].

Given adequate strength, a bushing may be sized to minimize the frictional torque on the shaft. This torque is given in N-m by:

$$T_f = \mu\,F\,\frac{D}{2}\,10^{-3} \qquad (7)$$

Programming

The problem of designing a bushing is now defined mathematically. The constants which specify the particular application are: 1) the radial load, F; 2) the shaft speed, Ω; and 3) the coefficient of friction, μ. The two design parameters to be selected are: 1) the bushing length, L; and 2) the shaft diameter, D. The three inequality constraints are: 1) the length to diameter

18

PRESSURE — P (MPa)

PV_{max} = 0.11 MPa—m/s

P_{max}

VELOCITY — V_s (m/s)
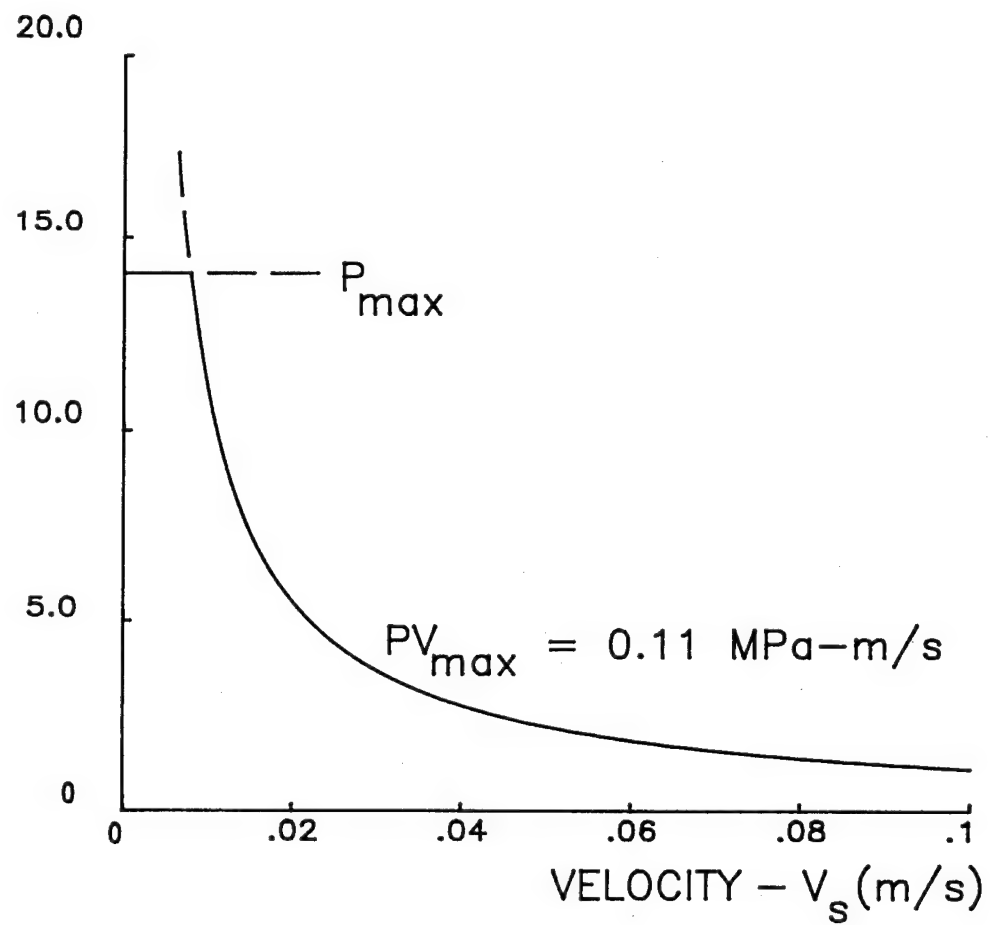
BUSHING CAPACITY LIMITS

FIGURE 1

ratio, β; 2) the acceptable pressure, P_{max}; and 3) the acceptable pressure times velocity factor, PV_{max}. All three constraints are upper bounds. The objective function, which is to be minimized, is the frictional torque, T_f. These quantities are summarized in Table 2.

The relations for the constraints are equations (1), (4) and (6), and the relation for the objective function is equation (7). A subroutine BOUNDS which is written to determine the constrained values using equations (1), (2) and (6) is listed in Table 3. And a subroutine VALUES which is written to determine the objective function value using equation (7) is listed in Table 4.

The simplicity of the subroutines matches the simplicity of the relations. In each subroutine, the input constant and design parameter vectors are converted to individual variables which have names that identify them more clearly. Then the equations are entered in an easily checked format and the results are transferred to the output constrained variable or objective function vectors. Note that the vector quantity subscripts match the input data order. These quantities are numbered in the output file echo of the input to assist the user in verifying that the proper input and output quantities are used for the equations in the two analysis subroutines.

Once written, these two subroutines must be compiled and linked to program SEEK to generate an executable program to perform the optimization. One way to do this is to add the subroutines to the end of the source file for program SEEK.FOR, save the combined program with a problem specific name such as BUSHING.FOR and compile it. The result will be an executable file, BUSHING.EXE. A second way would be to compile SEEK.FOR and the two analysis subroutines BOUNDS.FOR and VALUES.FOR separately to generate object files

20

Table 2

Bushing Optimization Parameters

Constants	Design Parameters	Inequality Constraints	Objective Function
F	L	$\dfrac{L}{D} \leq \beta_{max}$	$(T_f)_{min}$
Ω	D	$P \leq P_{max}$	
μ		$P\,V_s \leq PV_{max}$	

21

Table 3

Bushing Constraint Evaluation Subroutine Bounds

```
      SUBROUTINE BOUNDS(CONST,NCO,X,NX,VCSTR,NCS)
C
C               BOUNDS DETERMINES THE PRESENT CONSTRAINT
C                       FUNCTION VALUES
C
C         FOR A BUSHING DESIGN EXAMPLE
C
C       PARAMETERS:
C
C           CONST - FIXED DESIGN CONSTANT
C           NCO   - NUMBER OF DESIGN CONSTANTS
C           NCS   - NUMBER OF INEQUALITY CONSTRAINTS
C           NX    - NUMBER OF INDEPENDENT DESIGN PARAMETERS
C           VCSTR - PRESENT CONSTRAINT VALUES
C           X     - PRESENT DESIGN PARAMETER VALUES
C
C           ALL VALUES ARE IN PROBLEM UNITS
C
C           CONST(1) = F        - RADIAL LOAD (POUNDS)
C           CONST(2) = N        - SHAFT SPEED (RPM)
C           CONST(3) = f        - FRICTION COEFFICIENT
C
C           X(1) = L            - BUSHING LENGTH    (IN)
C           X(2) = D            - BUSHING DIAMETER (IN)
C
C           VCSTR(1) = P        - AVERAGE BUSHING CONTACT PRESSURE
C                                     (PSI)
C           VCSTR(2) = PV       - BUSHING PRESSURE TIMES VELOCITY
C                                     FACTOR (PSI - FT/MIN)
C           VCSTR(3) = L/D      - BUSHING LENGTH TO DIAMETER RATIO
C
C
      DIMENSION CONST(NCO),X(NX),VCSTR(NCS)
      PI = 3.14159265
      FORCE = CONST(1)
      RPM = CONST(2)
      BLEN = X(1)
      DIA = X(2)
      PRESS = FORCE/(BLEN*DIA)
      PV = 0.001*PI*FORCE*RPM/(60.0*BLEN)
      RATIO = BLEN/DIA
      VCSTR(1) = PRESS
      VCSTR(2) = PV
      VCSTR(3) = RATIO
      RETURN
      END
C
```

Table 4

Bushing Objective Function Evaluation Subroutine Values

```
      SUBROUTINE VALUES(CONST,NCO,X,NX,OBJECT,NOB)
C
C                  VALUES DETERMINES THE PRESENT DESIGN
C                        OBJECTIVE FUNCTION VALUES
C
C           FOR A BUSHING DESIGN EXAMPLE
C
C           PARAMETERS:
C
C                  CONST  - FIXED DESIGN CONSTANT
C                  NCO    - NUMBER OF DESIGN CONSTANTS
C                  NOB    - NUMBER OF OBJECTIVE FUNCTION TERMS
C                  NX     - NUMBER OF INDEPENDENT DESIGN VARIABLES
C                  OBJECT - PRESENT OBJECTIVE FUNCTION VALUES
C                  X      - PRESENT DESIGN PARAMETER VALUES
C
C           ALL VALUES ARE IN PROBLEM UNITS
C
C                  CONST(1) = F      - RADIAL LOAD (POUNDS)
C                  CONST(2) = N      - SHAFT SPEED (RPM)
C                  CONST(3) = f      - FRICTION COEFFICIENT
C
C                  X(1) = L          - BUSHING LENGTH   (IN)
C                  X(2) = D          - BUSHING DIAMETER (IN)
C
C                  OBJECT(1) = Tf    - BUSHING FRICTION TORQUE (LB - IN) C
C
C
      DIMENSION CONST(NCO),X(NX),OBJECT(NOB)
      FORCE = CONST(1)
      FRICT = CONST(3)
      DIA = X(2)
      TORQUE = 0.001*FRICT*FORCE*DIA/2.0
      OBJECT(1) = TORQUE
      RETURN
      END
C
```

only. These object files can then be linked with SEEK.OBJ listed first to produce an executable file, BUSHING.EXE [12,13].

Since this is a two parameter design problem with a single objective function, one can draw two graphs which illustrate the optimization. The first is called a design space in that it is a graph in coordinates which match the design parameters. Points in the graph represent specific design parameter values or designs. Plotted in this graph are the design constraint limits. These constraint limits divide the design space of potential designs into two regions: 1) an acceptable design region in which all design constraints are satisfied, and 2) an unacceptable design region in which at least one design constraint is violated. Figure 2 is a graph of a design space for the bushing design problem.

The second graph, Figure 3, is a plot of the objective function versus a design parameter. If the objective function were a function of both design parameters, a contour plot on the same coordinates as Figure 2 would be required to show how the objective function varies for different designs. Since the objective function of equation (7) is not a function of the bushing length, a simple graph of friction torque versus shaft diameter shows how this property varies for the potential designs. Figure 3 is drawn directly below the design space so that the objective function of friction torque can be visualized as a plane contour rising in the design space above.

On inspection of these graphs, it is obvious that the optimal design is the bushing with the smallest shaft diameter which satisfies the three inequality constraints plotted in Figure 2. Once this information is known, there is no need to go through a formal optimization to find the optimal design. Optimization techniques have their greatest value for problems for

24
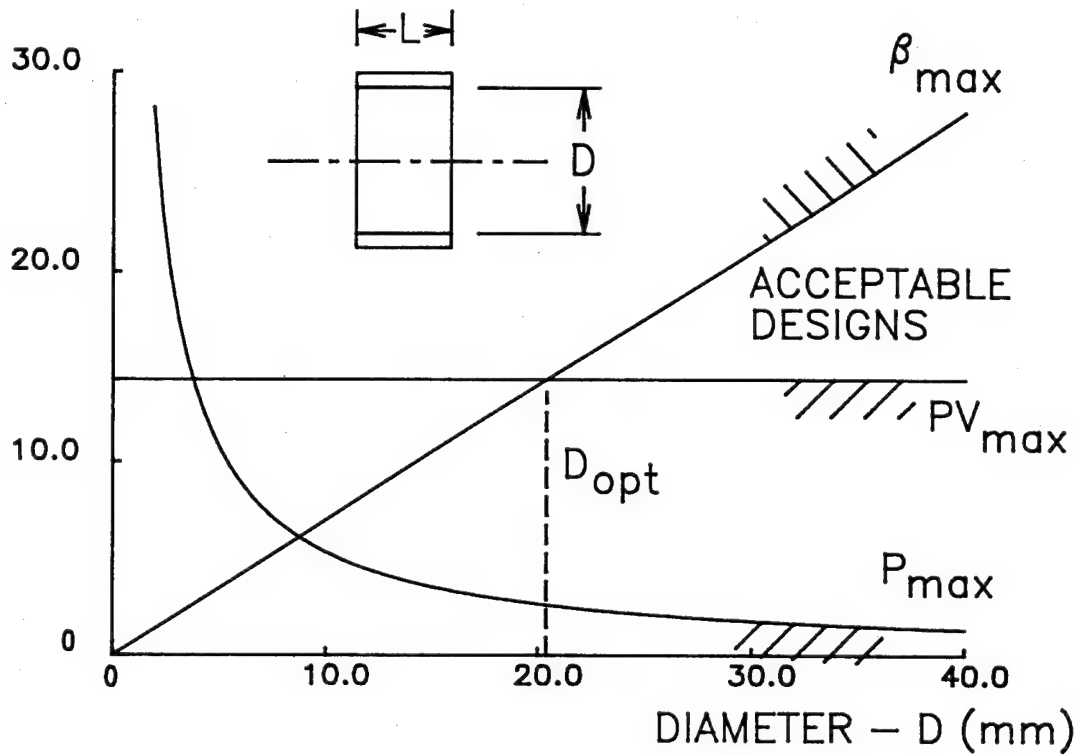
BUSHING DESIGN SPACE

FIGURE 2
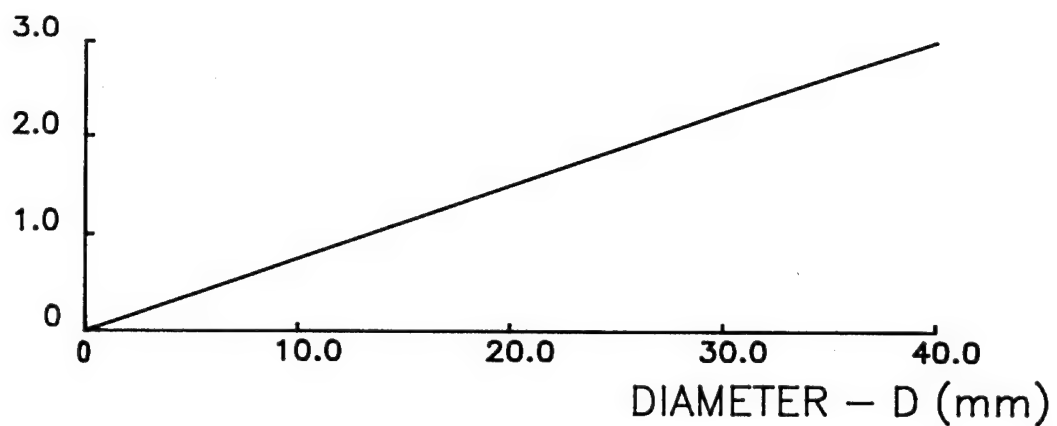
BUSHING OBJECTIVE FUNCTION

FIGURE 3

which optimum solutions are not yet known. Knowing this optimum will help us verify the effectiveness of the modified gradient method. But once an optimum solution is known, either from a graphical analysis or by a computer optimization, it is more efficient to calculate it directly [14].

Numerical Example

For an example, consider the design of a bushing to support 750 N at a shaft speed of 40 RPM. The shaft is steel and the bushing is to be nylon which has a coefficient of friction with steel of 0.2 and which has a design pressure limit of 14 MPa and a design PV limit of 0.11 MPa m/s. In this design, the length is to be limited to be less than or equal to seventy percent of the shaft diameter and the shaft size and bushing length are to be in whole mm's.

Table 5 is a listing of an input file for this problem, including line numbers which are not part of the input file. The first line is the problem title. The next line is the number of constants, 3, which is followed by three sets of three lines. Each constant is identified by its value, its name and its dimension.

The frictional coefficient has no dimension, so its dimension line is left blank in order for the following line which is the number of independent design parameters, 2, to appear in its proper place. If this line 11 were not left blank, an error checking routine would identify an error in the input file on line 13 and stop the program. Line 13 is the next line of text in the input file - the name of the first design variable. Due to the missing line 11, the program would try to read this text as the numerical value for the first design variable. Although this error message does not point directly to the cause of the reading error, it does indicate the presence of

26

Table 5

First Bushing Input File

Line	Column 1
1	RADIAL NYLON BUSHING
2	3
3	750.0
4	RADIAL LOAD
5	NEWTONS
6	40.0
7	SHAFT SPEED
8	RPM
9	0.2
10	FRICTION COEFFICIENT
11	
12	2
13	0.0 10.0 5.0
14	BUSHING LENGTH
15	mm
16	0.0 10.0 5.0
17	BUSHING DIAMETER
18	mm
19	3
20	UPPER 14.0
21	CONTACT PRESSURE
22	MPa
23	UPPER 0.11
24	PV FACTOR
25	MPa – m/s
26	UPPER 0.7
27	LENGTH TO DIAMETER
28	RATIO
29	MIN
30	1
31	1.0
32	FRICTION TORQUE
33	N – m

an error. Checking the data on line 13 and the lines that precede it should lead to this discovery in a short amount of time.

The next six lines contain the design parameter values, names and dimensions. The value lines contain three numbers: 1) the low estimate, 2) the high estimate, and 3) the initial estimate. At this point in the solution, we know the least about the values to enter for these design parameters. Let us guess ranges from zero to ten mm and initial values of five mm for the two design variables. The next line contains the number of design constraints, 3. Following are nine lines with the three constraint limit types and values with their decimal points on the first lines, their names on the second lines and their units on the third lines.

The data for the objective function vector follows. The next line contains the letters 'MIN' to identify minimization as the direction of optimization. This is followed by a line with the single value of one to indicate that the objective function has only one term. The last three lines are the weighting coefficient value and the name and units for the term.

This file is saved with a name such as NYLON1.IN. The compiled program BUSHING.EXE can now be run by typing BUSHING at the prompt. As shown in Figure 4, the program will request the prefix for an input file, which should be NYLON1 in this case. Since "data points" is not in the first constant description, the program will bypass this option and not try to open and read a data file. After receiving the input, the program will run. It will echo the input data to the screen, with several PAUSEs. Press the ENTER key at each PAUSE to continue execution of the program. For the input data of Table 5, it turns out that the initial guess is too small and the ranges are small also. The program will try to change this initial guess into a design

```
D:\>BUSHING

    ENTER INPUT FILE NAME PREFIX FOR A .IN ASCII FILE.
    IT MAY INCLUDE A PATH TO ANOTHER DIRECTORY,
    WITH NO MORE THAN 26 CHARACTERS.

    IF THE FIRST CONSTANT IS THE NUMBER OF DATA POINTS,
    AND ITS DESCRIPTION INCLUDES "DATA POINTS."
    A DATA FILE WITH X,Y DATA PAIRS AND A .DAT
    EXTENSION SHOULD ALSO EXIST IN THE SAME DIRECTORY.

    THE OUTPUT FILE WILL HAVE THE SAME PREFIX WITH A .OUT EXTENSION.

NYLON1
```

Seek Screen Image at Start of Program

FIGURE 4

which does satisfy the design constraints, but since the design ranges are small also, the improvement steps are too small to reach the acceptable design space in the twenty steps allowed by the program.

Table 6 is a listing of the output file for this trial. In this output, one can see that the shaft diameter was tripled in an attempt to reach the region of good designs. The PV limit was the major constraint, with values twenty percent higher than the limit for the revised design and three times the limit for the initial design. A second input file, NYLON2.IN can now be made by copying the first and modifying the design parameter initial values. In this second file the ranges of both design parameters are left at zero to ten mm and the initial values are increased to twenty mm for the length and thirty mm for the diameter. These are the only changes from the first input file. Table 7 lists the new input file, NYLON2.IN.

The results of running the program again with the new input file are listed in Table 8. An optimum design was found in 45 steps with a length of 14.28 mm and a diameter of 20.41 mm. The design has a friction torque of 1.53 N-m and satisfies all three constraints. The two limiting constraints which are just satisfied are the PV factor limit and the length to diameter ratio. These results are consistent with the graphical results of Figures 2 and 3.

Note that the input ranges of the design variables only set the sensitivity of the search, they do not limit the design to have diameters less than 10 mm. To make that a limit, one must add an upper limit of 10 mm to the diameter as a fourth constraint. This would prevent the optimizer from finding a solution, since no design with a diameter less than 10 mm can have a

Table 6

First Bushing Output Log File

RADIAL NYLON BUSHING

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1	RADIAL LOAD	750.00000	NEWTONS
2	SHAFT SPEED	40.00000	RPM
3	FRICTION COEFFICIENT	0.20000	

THERE ARE 2 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

	LOW	HIGH	INITIAL	
1 BUSHING LENGTH	0.00000	10.00000	5.00000	mm
2 BUSHING DIAMETER	0.00000	10.00000	5.00000	mm

THE 3 CONSTRAINT LIMITS ARE:

				TYPE
1	CONTACT PRESSURE	14.00000	MPa	UPPER
2	PV FACTOR	0.11000	MPa – m/s	UPPER
3	LENGTH TO DIAMETER	0.70000	RATIO	UPPER

MINIMIZE THE OBJECTIVE FUNCTION.

OBJ = FRICTION TORQUE   IN N – m   TIMES   1.0000

Table 6 Continued

<u>First Bushing Output Log File</u>

OPTIMIZATION COULD NOT BEGIN - BAD INITIAL VALUE.

THE INITIAL VALUE FOR THE PROBLEM VIOLATED
AT LEAST ONE INEQUALITY CONSTRAINT.

THE PROGRAM COULD NOT FIND ANOTHER VECTOR WHICH
SATISFIED ALL THE INEQUALITY CONSTRAINTS.

		X INITIAL	X MODIFIED	
1	BUSHING LENGTH	5.00000	12.07892	mm
2	BUSHING DIAMETER	5.00000	16.83625	mm

THE 3 CONSTRAINT VALUES FOR X INITIAL ARE:

					LIMIT	TYPE
1	CONTACT PRESSURE	=	30.000	MPa	14.000	UPPER
2	PV FACTOR	=	.31416	MPa - m/s	.11000	UPPER
3	LENGTH TO DIAMETER	=	1.0000	RATIO	.70000	UPPER

THE 3 CONSTRAINT VALUES FOR X MODIFIED ARE:

					LIMIT	TYPE
1	CONTACT PRESSURE	=	3.6880	MPa	14.000	UPPER
2	PV FACTOR	=	.13004	MPa - m/s	.11000	UPPER
3	LENGTH TO DIAMETER	=	.71744	RATIO	.70000	UPPER

Table 7

Second Bushing Input File

```
RADIAL NYLON BUSHING
3
750.0
RADIAL LOAD
NEWTONS
40.0
SHAFT SPEED
RPM
0.2
FRICTION COEFFICIENT

2
0.0   10.0   20.0
BUSHING LENGTH
mm
0.0   10.0   30.0
BUSHING DIAMETER
mm
3
UPPER   14.0
CONTACT PRESSURE
MPa
UPPER   0.11
PV FACTOR
MPa - m/s
UPPER   0.7
LENGTH TO DIAMETER
RATIO
MIN
1
1.0
FRICTION TORQUE
N - m
```

Table 8

Second Bushing Output Log File

RADIAL NYLON BUSHING

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1   RADIAL LOAD   750.00000   NEWTONS
2   SHAFT SPEED   40.00000   RPM
3   FRICTION COEFFICIENT   0.20000

THERE ARE 2 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

		LOW	HIGH	INITIAL	
1	BUSHING LENGTH	0.00000	10.00000	20.00000	mm
2	BUSHING DIAMETER	0.00000	10.00000	30.00000	mm

THE 3 CONSTRAINT LIMITS ARE:

TYPE

1	CONTACT PRESSURE	14.00000	MPa	UPPER
2	PV FACTOR	0.11000	MPa - m/s	UPPER
3	LENGTH TO DIAMETER	0.70000	RATIO	UPPER

MINIMIZE THE OBJECTIVE FUNCTION.

OBJ = FRICTION TORQUE   IN N - m   TIMES   1.0000

OPTIMIZATION SUCCESSFUL IN   45 STEPS

THE FINAL DESIGN VECTOR IS:

X(I)

1	BUSHING LENGTH	14.28342	mm
2	BUSHING DIAMETER	20.40649	mm

THE MINIMUM OBJECTIVE FUNCTION =   1.53049   , ITS COMPONENTS ARE:

1   FRICTION TORQUE   = 1.5305   N - m   TIMES   1.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION   = -0.677109E-04
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE = 0.390625E-03

Table 8 Continued

Second Bushing Output Log File

THE   3 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 CONTACT PRESSURE	= 2.5735	MPa	14.000	UPPER
2 PV FACTOR	= .10999	MPa – m/s	.11000	UPPER
3 LENGTH TO DIAMETER	= .69989	RATIO	.70000	UPPER

DESIGN CHECK

X(I)

1	BUSHING LENGTH	15.00000	mm
2	BUSHING DIAMETER	21.00000	mm

THE MINIMUM OBJECTIVE FUNCTION =   1.57500   , ITS COMPONENTS ARE:

1	FRICTION TORQUE	= 1.5750	N – m	TIMES	1.0000

THE   3 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 CONTACT PRESSURE	= 2.3810	MPa	14.000	UPPER
2 PV FACTOR	= .10472	MPa – m/s	.11000	UPPER
3 LENGTH TO DIAMETER	= .71429	RATIO	.70000	UPPER

DESIGN CHECK

X(I)

1	BUSHING LENGTH	15.00000	mm
2	BUSHING DIAMETER	22.00000	mm

THE MINIMUM OBJECTIVE FUNCTION =   1.65000   , ITS COMPONENTS ARE:

1	FRICTION TORQUE	= 1.6500	N – m	TIMES	1.0000

THE   3 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 CONTACT PRESSURE	= 2.2727	MPa	14.000	UPPER
2 PV FACTOR	= .10472	MPa – m/s	.11000	UPPER
3 LENGTH TO DIAMETER	= .68182	RATIO	.70000	UPPER

35

PV factor less than 0.11 MPa - m/s and a β less than 0.7. Repeated trials such as the first one would tell us that.

However, another constraint on the solution was that the diameter and length be in whole mm's. This can be obtained with the design check provision of the program, which is shown in Table 8. Once the numerical optimum has been found, the program lists the number of optimizing steps followed by: the found design parameter values for bushing length and bushing diameter, the objective function value and the three constrained variable values and limits. Then the program re-lists the design variables with their found values and offers the user the option to change them for a design check. Figure 5 shows this interaction. The user responded with a 'Y' to the question on trying another design and entered the two values of '15' and '21' for the bushing length and diameter. The program then printed the results to the screen and added them to the output file as shown in Table 8. This option is offered to the user at the end of each analysis until the response to the first question is 'N' which tells the program to close the output file and stop the program.

Increasing the length to 15 mm and the diameter to 21 mm increases the friction torque slightly to 1.575 N-m but still does not satisfy the length to diameter constraint. A second trial with a 15 mm length and a 22 mm diameter has a friction torque of 1.65 mm and satisfies all three modeled constraints and the additional requirement of standard sizes. This trial is the optimal design and is shown in Figure 6.

```
       FOR THE LAST DESIGN, THE DESIGN VARIABLE VECTOR IS:

                                  X(I)

  1  BUSHING LENGTH                   14.28342  mm
  2  BUSHING DIAMETER                 20.40649  mm


       DO YOU WISH TO MODIFY THE DESIGN AND CHECK IT ? (Y/N)

Y



       ENTER NEW VALUES FOR THE DESIGN VARIABLE VECTOR, X.
       ENTER A COMMA "," TO KEEP THIS VALUE.


  OLD BUSHING LENGTH            =        14.28342  mm          ,
  NEW BUSHING LENGTH            = ?

15

  OLD BUSHING DIAMETER          =        20.40649  mm          ,
  NEW BUSHING DIAMETER          = ?

21
```

Seek Design Verification Request Screen

FIGURE 5

22 mm

15 mm
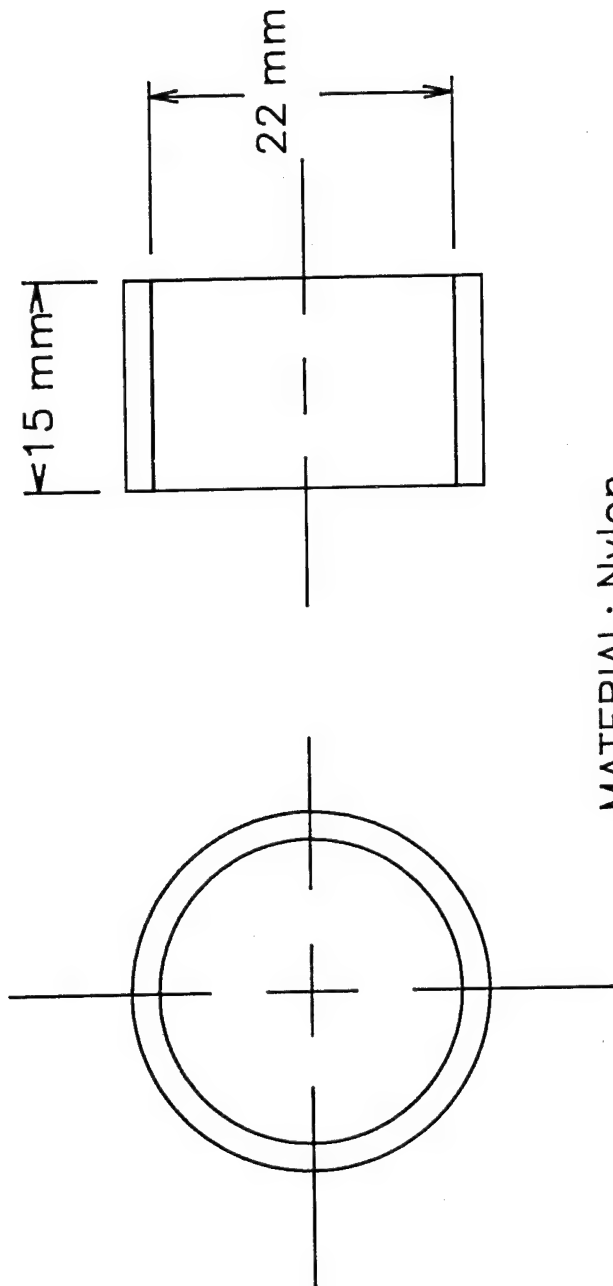
MATERIAL: Nylon
SPEED: 40 RPM
LOAD: 750 N

OPTIMUM BUSHING DESIGN

FIGURE 6

SPRING OPTIMIZATION

In this example, consider the design of a steel helical coil compression spring to support a varying load with a specified spring rate. The applied load varies between minimum and maximum values in service and may be larger if the spring is compressed solid at assembly. For a given spring material, four geometric parameters define the spring: 1) the wire diameter, d_w; 2) the mean coil diameter, D; 3) the number of active coils, N_a; and 4) the height of the spring when unloaded, h_f.

In this design problem, one more requirement can be placed on the performance of the spring: it could have a specified outside diameter, or work over a rod of a given diameter or it could have a required height under load. Instead, we will let the optimizer find a spring with a minimized property which can support the specified loads with the given spring rate. Three separate objective functions will be minimized: 1) spring weight, 2) spring height, and 3) spring coil volume. The resulting designs will satisfy the load and deflection requirements, but they will be different.

For the spring design to have some practicality, the wire size should be selected from a finite list of available diameters and the mean coil diameter should also be a standard size.

Theory

A spring's performance is modeled by its strength and deflection. A helical coil spring supports its axial load as a torsional shear stress in the wire with a small additional direct shear stress. Figure 7 shows the axial load and the internal wire torque and shear which support it. Due to the curvature of the wire, Wahl determined a stress concentration factor, K_w, which compares the maximum shear stress in the wire to the nominal torsional
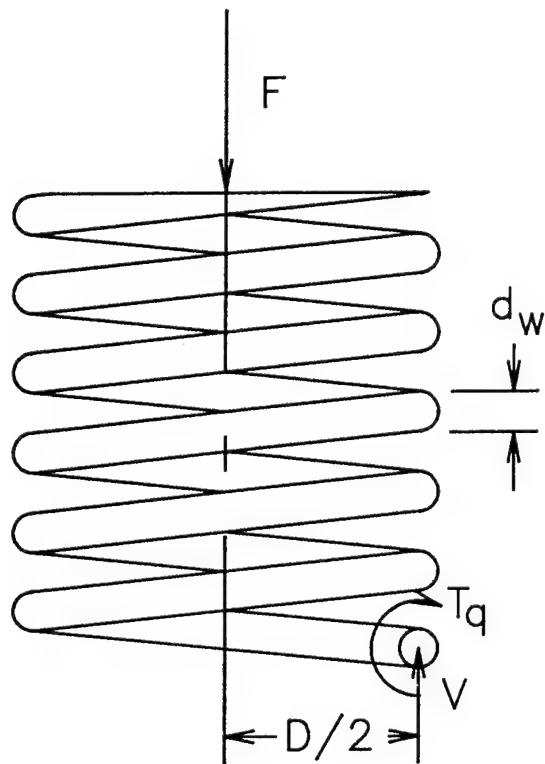
INTERNAL LOADS
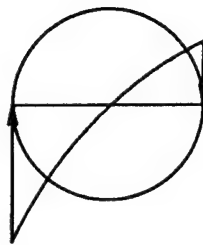
SHEAR STRESSES

HELICAL COIL SHEAR LOADS AND STRESSES

FIGURE 7

stress caused by the couple of the axial load through the center of the spring and the supporting shear force through the center of the wire [10,15]. The shear stress and stress concentration factor are:

$$\tau = \frac{K_w T_q D}{2.0 J} = \frac{8.0 K_w F D}{\pi d_w^3} = \frac{8.0 K_w F C}{\pi d_w^2} \tag{8}$$

and

$$K_w = \frac{4.0 C - 1.0}{4.0 C - 4.0} + \frac{0.615}{C} \tag{9}$$

where the spring index, C is:

$$C = \frac{D}{d_w} \tag{10}$$

The deflection of a helical coil spring at any load, F, is a result of the twist in the wire due to the applied torque, T_q:

$$\delta = \frac{D}{2} \theta = \frac{D}{2} \frac{T_q L}{J G} \tag{11}$$

or

$$\delta = \frac{D/2 \ (F \ D/2 \ ) \ (\pi D N_a)}{(\pi d_w^4/32 \ ) G} = \frac{8 F D^3 N_a}{d_w^4 G} \tag{12}$$

or

$$\delta = \frac{8 F C^3 N_a}{d_w G} \tag{13}$$

The stiffness of the spring or its rate is thus:

$$k = \frac{F}{\delta} = \frac{d_w G}{8.0 C^3 N_a} \tag{14}$$

41

These basic equations for stress (8), deflection (13) and stiffness (14) can be applied to the specified loads to model the performance of the spring. To avoid yielding, the maximum stress in the spring should be less than the shear yield strength of the wire divided by the design stress, or the static design factor, N_s, should be greater than the desired design factor, where:

$$N_s = \frac{S_{sy}}{\tau_{sol}} \tag{15}$$

and the maximum stress in the spring, τ_{sol}, is found by using the solid height force, F_{sol}, in equation (8). The shear yield strength of the wire can be estimated as eighty percent of the ultimate shear strength of the wire. As spring wire is drawn to a smaller diameter, its ultimate strength increases. Using the octahedral shear factor of 0.577, the ultimate shear strength can be estimated as a function of the drawn wire size as [15]:

$$S_{su} = 0.577 \, S_{uc} \, d_w^a \tag{16}$$

In addition to having adequate strength to avoid yielding when compressed solid, springs should be able to support an infinite number of loading cycles without experiencing a fatigue failure. Figure 8 is a Soderberg diagram of alternating shear stress versus mean shear stress, which shows the reduction in alternating strength with increasing mean stress, for a helical coil compression spring. Since the stress concentration affects the alternating stress directly and does not affect the mean stress influence, the Wahl factor is applied only to the alternating shear stress. An algebraic trick, which allows all stresses to be calculated with equation (8), is to multiply the stress concentration factor by the ultimate strength to divide it out of the mean stress.

42

ALTERNATING SHEAR STRESS τ_a (MPa)

LOAD LINE

GOODMAN LIMIT

S_{se}

$\dfrac{S_{se}}{N_f}$

0

$K_w S_{su}$

$\dfrac{K_w S_{su}}{N_f}$

MEAN SHEAR STRESS τ_m (MPa)
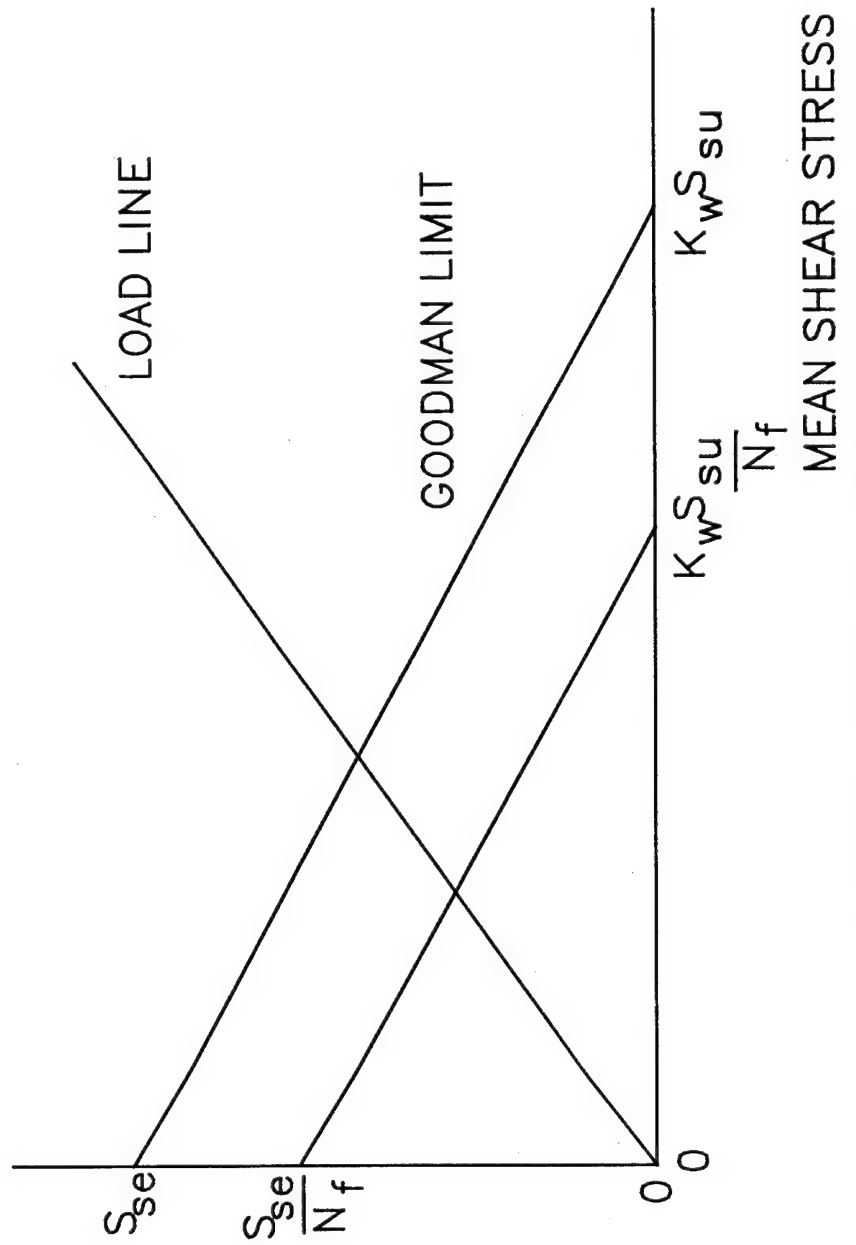
SPRING FATIGUE STRENGTH

FIGURE 8

43

In Figure 8, the negative sloped lines show the Goodman criteria for fatigue strength with and without the design factor, N_f, and the positive sloped line shows the load ratio of alternating stress, τ_a, to mean stress, τ_m, for this application. The design factor equation for this criteria is [15]:

$$\frac{1}{N_f} = \frac{\tau_m}{K_w \, S_{su}} + \frac{\tau_a}{S_{se}} \qquad (17)$$

or

$$N_f = \frac{1.0}{\dfrac{\tau_m}{K_w \, S_{su}} + \dfrac{\tau_a}{S_{se}}} \qquad (18)$$

The mean stress is calculated with equation (8) using the average applied load:

$$F_m = \frac{F_{max} + F_{min}}{2.0} \qquad (19)$$

And the alternating stress is calculated using the alternating applied load:

$$F_a = \frac{F_{max} - F_{min}}{2.0} \qquad (20)$$

Unlike the ultimate wire strength, the fatigue strength of the steel wire, S_{se}, is constant at these high strengths.

Figure 9 is a plot of the force on the spring versus the height of the spring. Near the solid height, the axial load on the spring increases rapidly as the coils close. In service, the spring operates between the working heights of h_{min} and h_{max} with loads of F_{max} and F_{min}. The unloaded height of the spring is the sum of the deflection to solid height and the solid height:

44
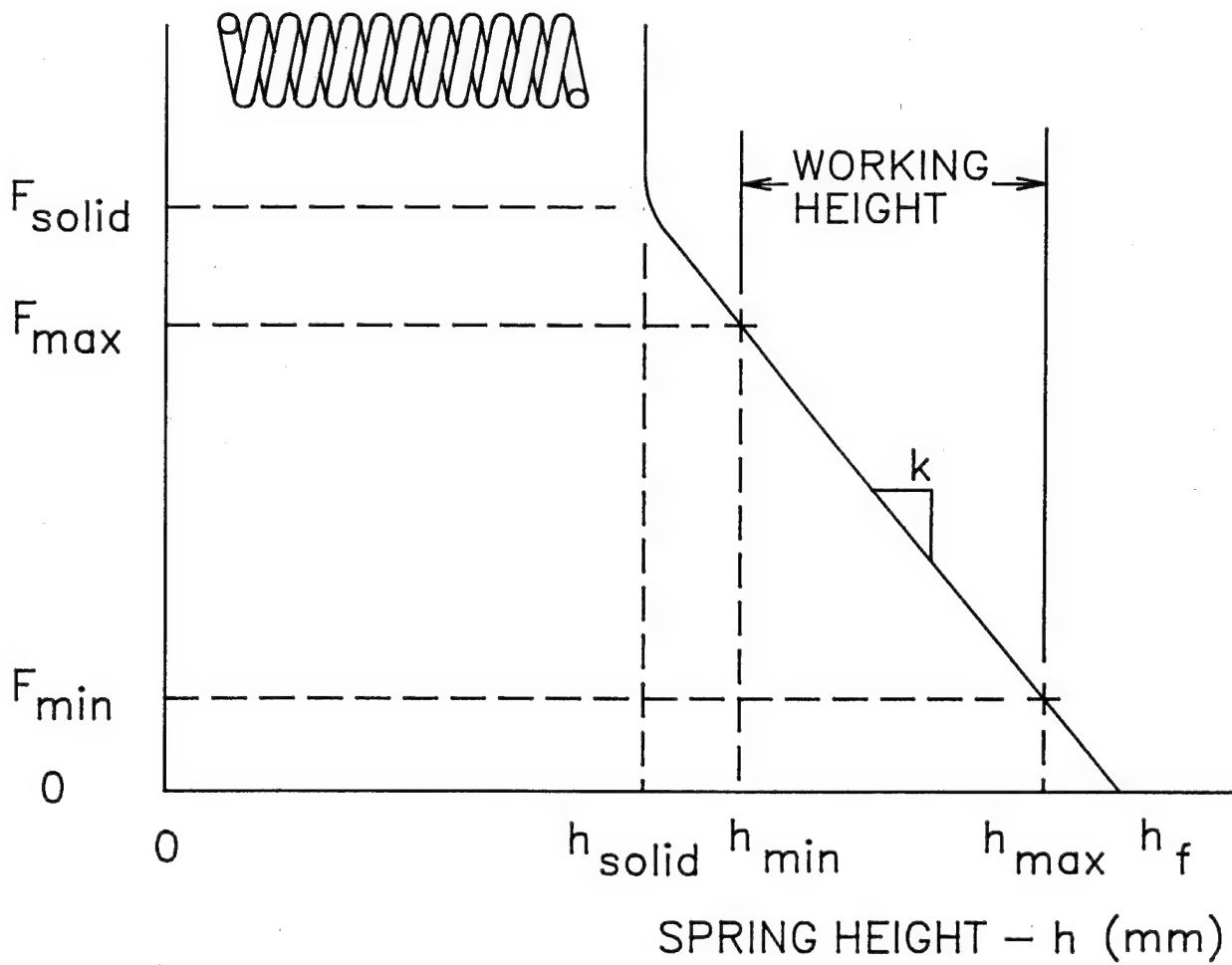
SPRING FORCE VERSUS POSITION

FIGURE 9

45

$$h_f = \frac{F_{sol}}{k} + 1.01 \, d_w \, (N_a + N_e) \qquad (21)$$

Coil weight is a direct function of the volume of wire in the spring:

$$V_w = \frac{\pi}{4} \, d_w^2 \, (N_a + N_e) \, \pi D \qquad (22)$$

with:

$$W_t = w \, V_w \qquad (23)$$

The volume of the coil is the area of the outside diameter's circular disc times the spring height:

$$V_{coil} = \frac{\pi}{4} \, OD^2 \, h_f \qquad (24)$$

Programming

Table 9 summarizes the design problem in terms of the constants which define the problem, the design parameters which are to be found, the equality and inequality constraints on the design and the three separate objective functions which will be sought. In the initial formulation, there are two equality constraints on the stiffness and the force at solid height, which can be used to reduce the number of independent design parameters from four to two. There are four active inequality constraints: 1) that the number of active coils be positive, 2) that the fatigue design factor be greater than the desired design factor, 3) that the static design factor at assembly be greater than the desired design factor, and 4) that the spring index be greater than two, so a coil can be manufactured.

Table 9

Initial Spring Optimization Parameters

Constants	Design Parameters	Constraints Equality	Inequality	Objective Function
F_{min} F_{max} F_{solid} k N_e N_{des} S_{se} S_{su} G w	d_w D N_a h_f	k F_{solid}	$N_a > 0.0$ $N_f > N_{des}$ $N_s > N_{des}$ $C > 2.0$	$(h_f)_{min}$ or $(W_t)_{min}$ or $(Vol)_{min}$

Table 10

Revised Spring Optimization Parameters

Constants	Design Parameters	Inequality Constraints	Objective Function
F_{min}	d_w	$N_a > 0.0$	$(h_f)_{min}$
F_{max}	D	$N_f > N_{des}$	or
F_{solid}		$N_s > N_{des}$	$(W_t)_{min}$
k		$C > 2.0$	or
N_e		$\tau_{max} > 0.0$	$(Vol)_{min}$
S_{se}		$OD > 0.0$	
S_{uc}		$V_w > 0.0$	
a			
G			
w			

From the stiffness equation (14), one can relate the number of active coils to the wire and mean coil diameters, the stiffness and the shear modulus of the material:

$$N_a = \frac{d_w \, G}{8.0 \, k \, C^3} \tag{25}$$

Equation (21) relates the spring height to: the maximum force, F_{sol}; the spring rate; the wire diameter; and the numbers of active and end coils. These two equations convert two design parameters from independent parameters to dependent parameters and simplify the optimization. Table 10 is a second pass at formulating the optimization problem in this simpler form. Three inactive constraints have been added to the inequality constraint list to make the optimizer report these properties. All properties are constrained to be greater than zero, which they will be for all designs. The watched properties are: 1) maximum static stress, τ_{max}; 2) outside diameter, OD; and 3) wire volume, V_w. The final differences between the two formulations are in the constant list: 1) the elimination of the design factor from the constant list since it is used in the constraint limit list and, 2) the replacement of the ultimate shear strength of the wire by the ultimate tensile strength constant and the wire power to enable the program to vary the strengths with wire size.

Figure 10 is a plot of the design space for this reformulated optimization. The graph plots the two independent design parameters: the wire diameter, d_w, and the mean coil diameter, D, versus each other. On the graph are drawn constraint lines for the four active constraints. The number of active coils constraint, N_a, is drawn for a limit of 0.1 coils to show its shape. A limit of zero coils lies on the x axis. Designs which satisfy all

48

WIRE DIAMETER – d_w (mm)

8.0

6.0

4.0

2.0

0

C

ACCEPTABLE DESIGNS

SOLID

FATIGUE

N_a

0   50.0   100.0   150.0   200.0   250.0

MEAN COIL DIAMETER – D (mm)
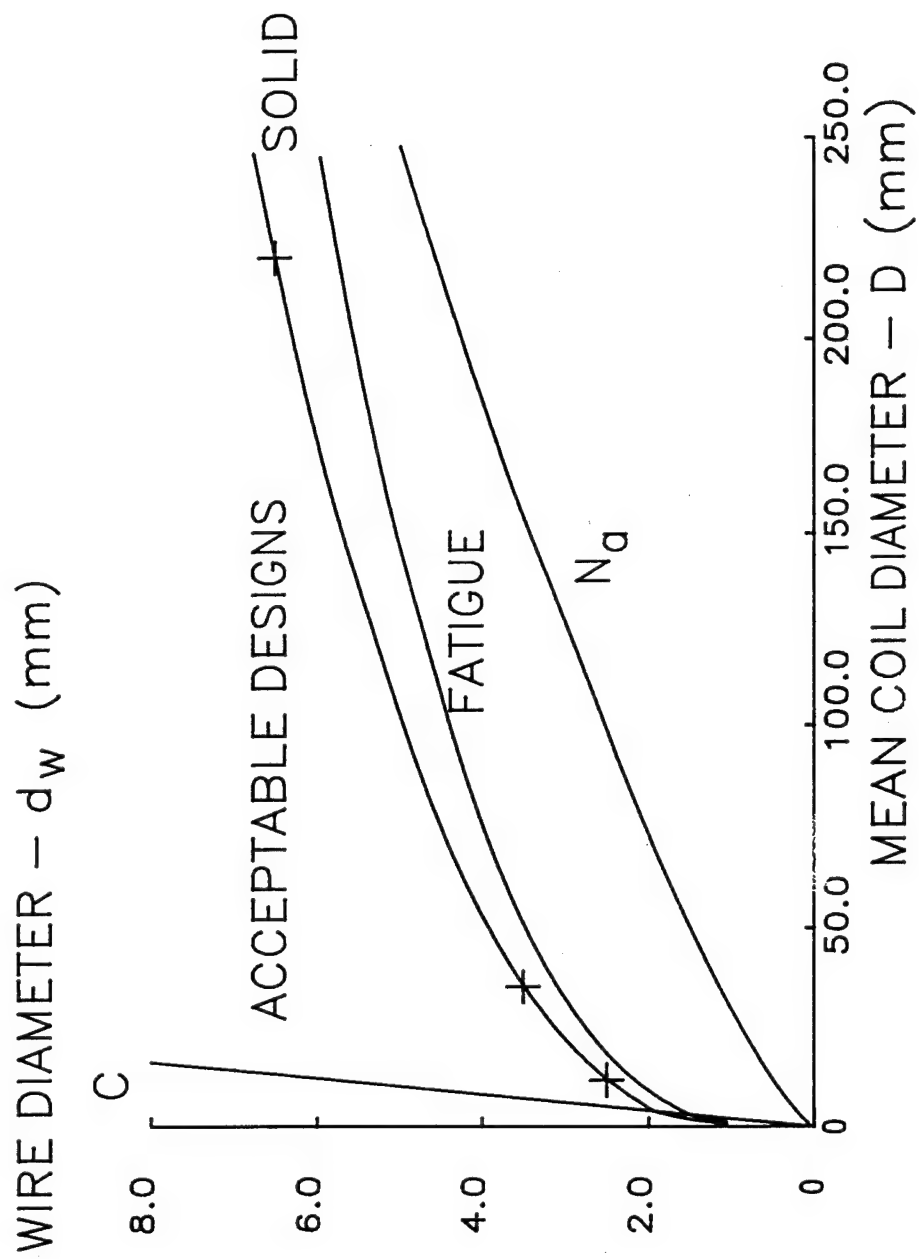
HELICAL COIL SPRING DESIGN SPACE

FIGURE 10

49

constraints can be found in the region labeled acceptable designs which is between the spring index limit, labeled C, and the solid stress limit, labeled SOLID.

Each of the three objective functions are complex functions of these parameters, so a single plot of the objective function versus mean coil diameter is not possible. Contour plots for each objective function would have to be drawn on the design space to obtain a graphical solution to the problem.

Two analysis subroutines must now be written to operate on the constants and design parameters of Table 10 and determine the constraint values and the objective function values listed. Subroutine BOUNDS, which is listed in Table 11, takes the constant array and the design parameter array from the calling list and determines the constraint values. As with the bushing example, this is done in three steps to clarify the calculations: 1) conversion of input arrays to variables, 2) calculation of the properties and 3) transferring the property values to the constraint property array. The analysis of equations (8) through (25) is used in the subroutine. Subroutine VALUES, which is listed in Table 12, performs a similar determination of the three objective function values. Once written, these two subroutines must be compiled and linked to program SEEK to generate an executable program to perform the optimization.

Numerical Example

For the example, consider the design of a spring to have a spring rate of 4 kN/m. In use, the spring will see a load which varies from a minimum of 24 N to a maximum of 120 N. The spring should not go solid until the applied load reaches a value of 150 N. The spring wire is to be selected from stock

Table 11

Spring Constraint Evaluation Subroutine Bounds

```
      SUBROUTINE BOUNDS(CONST,NCO,X,NX,VCSTR,NCS)
C
C                  BOUNDS DETERMINES THE PRESENT CONSTRAINT
C                           FUNCTION VALUES
C
C              FOR A HELICAL COIL SPRING DESIGN
C
C         PARAMETERS:
C
C                  CONST - FIXED DESIGN CONSTANT
C                  NCO   - NUMBER OF DESIGN CONSTANTS
C                  NCS   - NUMBER OF INEQUALITY CONSTRAINTS
C                  NX    - NUMBER OF INDEPENDENT DESIGN VARIABLES
C                  VCSTR - PRESENT CONSTRAINT VALUES
C                  X     - PRESENT DESIGN VARIABLE VALUES
C
C                  ALL VALUES ARE IN PROBLEM UNITS
C
C                  CONST(1)  = Fmin  - MINIMUM FORCE (NEWTONS)
C                  CONST(2)  = Fmax  - MAXIMUM FORCE (NEWTONS)
C                  CONST(3)  = Fsol  - MINIMUM SOLID FORCE (NEWTONS)
C                  CONST(4)  = k     - SPRING RATE (kN/m)
C                  CONST(5)  = Ne    - NUMBER OF DEAD COILS
C                  CONST(6)  = Sse   - SHEAR FATIGUE STRENGTH (MPa)
C                  CONST(7)  = Suc   - TENSILE STRENGTH COEFICIENT (MPa)
C                  CONST(8)  = a     - TENSILE STRENGTH WIRE POWER
C                  CONST(9)  = G     - SHEAR MODULUS (MPa)
C                  CONST(10) = w     - WEIGHT DENSITY (kN/m**3)
C
C                  X(1) = dw          - WIRE DIAMETER (mm)
C                  X(2) = D           - MEAN COIL DIAMETER (mm)
C
C                  VCSTR(1) = Na      - NUMBER OF ACTIVE COILS
C                  VCSTR(2) = Nf      - FATIGUE DESIGN FACTOR
C                  VCSTR(3) = Ns      - STATIC DESIGN FACTOR
C                  VCSTR(4) = C       - SPRING INDEX
C                  VCSTR(5) = Tmax    - MAXIMUM SHEAR STRESS (MPa)
C                  VCSTR(6) = OD      - OUTSIDE DIAMETER (mm)
C                  VCSTR(7) = Vw      - SPRING WIRE VOLUME (mm**3)
C
C
```

Table 11 Continued

Spring Constraint Evaluation Subroutine Bounds

```
DIMENSION CONST(NCO),X(NX),VCSTR(NCS)
PI = 3.14159265
FMIN = CONST(1)
FMAX = CONST(2)
FSOL = CONST(3)
RATE = CONST(4)
ZNE = CONST(5)
SSE = CONST(6)
CSUT = CONST(7)
ASUT = CONST(8)
G = CONST(9)
DW = X(1)
D = X(2)
C = D/DW
ZNA = DW*G/(8.0*C*C*C*RATE)
SSU = 0.577 * CSUT * DW**ASUT
SSY = 0.8*SSU
FA = (FMAX - FMIN)/2.0
FM = (FMAX + FMIN)/2.0
SKW = (4.0*C - 1.0)/(4.0*C - 4.0) + 0.615/C
TA = (8.0*SKW*FA*C/(PI*DW*DW))
TM = (8.0*FM*C/(PI*DW*DW))
ZNF = 1.0/(TM/SSU + TA/SSE)
TMAX = (8.0*SKW*FSOL*C/(PI*DW*DW))
ZNS = SSY/TMAX
OD = D + DW
ZNT = ZNA + ZNE
VW = 0.25*PI*DW*DW*ZNT*PI*D
VCSTR(1) = ZNA
VCSTR(2) = ZNF
VCSTR(3) = ZNS
VCSTR(4) = C
VCSTR(5) = TMAX
VCSTR(6) = OD
VCSTR(7) = VW
RETURN
END
C
```

Table 12

Spring Objective Function Evaluation Subroutine Values


```
      SUBROUTINE VALUES(CONST,NCO,X,NX,OBJECT,NOB)
C
C              VALUES DETERMINES THE PRESENT DESIGN
C                    OBJECTIVE FUNCTION VALUES
C
C         FOR A CANTILEVER BEAM DESIGN EXAMPLE
C
C      PARAMETERS:
C
C           CONST  - FIXED DESIGN CONSTANT
C           NCO    - NUMBER OF DESIGN CONSTANTS
C           NOB    - NUMBER OF OBJECTIVE FUNCTION TERMS
C           NX     - NUMBER OF INDEPENDENT DESIGN VARIABLES
C           OBJECT - PRESENT OBJECTIVE FUNCTION VALUES
C           X      - PRESENT DESIGN VARIABLE VALUES
C
C           ALL VALUES ARE IN PROBLEM UNITS
C
C           CONST(1)  = Fmin  - MINIMUM FORCE (NEWTONS)
C           CONST(2)  = Fmax  - MAXIMUM FORCE (NEWTONS)
C           CONST(3)  = Fsol  - MINIMUM SOLID FORCE (NEWTONS)
C           CONST(4)  = k     - SPRING RATE (kN/m)
C           CONST(5)  = Ne    - NUMBER OF DEAD COILS
C           CONST(6)  = Sse   - SHEAR FATIGUE STRENGTH (MPa)
C           CONST(7)  = Suc   - TENSILE STRENGTH COEFICIENT (MPa)
C           CONST(8)  = a     - TENSILE STRENGTH WIRE POWER
C           CONST(9)  = G     - SHEAR MODULUS (MPa)
C           CONST(10) = w     - WEIGHT DENSITY (kN/m**3)
C
C           X(1) = dw         - WIRE DIAMETER (mm)
C           X(2) = D          - MEAN COIL DIAMETER (mm)
C
C           OBJECT(1) = Wt    - SPRING WEIGHT (NEWTONS)
C           OBJECT(2) = hf    - SPRING HEIGHT (mm)
C           OBJECT(3) = Vc    - SPRING CYLINDER VOLUME (mm**3)
C
C
      DIMENSION CONST(NCO),X(NX),OBJECT(NOB)
      PI = 3.14159265
      FSOL = CONST(3)
      RATE = CONST(4)
      ZNE = CONST(5)
      G = CONST(9)
      W = CONST(10)/1000000.0
```

Table 12 Continued

Spring Objective Function Evaluation Subroutine Values

```
DSOL = FSOL/RATE
DW = X(1)
D = X(2)
C = D/DW
OD = D + DW
ZNA = DW*G/(8.0*C*C*C*RATE)
ZNT = ZNA + ZNE
VW = 0.25*PI*DW*DW*ZNT*PI*D
WT = VW*W
HF = DSOL + DW*ZNT*1.01
VC = 0.25*PI*OD*OD*HF
OBJECT(1) = WT
OBJECT(2) = HF
OBJECT(3) = VC
RETURN
END
```
C

sizes which have whole and half mm values. The mean coil diameter should also have values with whole or half mm precision. The springs are to be made of hard drawn spring wire with an ultimate strength constant of 1510 MPa, a wire diameter strength variation exponent of -0.201, and a shear fatigue strength of 465 MPa. The acceptable design factor is 1.5 and the spring ends are to be squared and ground with one inactive coil at each end. The shear modulus of steel is 79,000 MPa and its weight density is 76.5 kN/m^3.

Table 13 is a listing of the input file for the minimum weight design option. The file begins with the title for the output file. The second line has the number of constants to follow - ten. The next thirty lines contain these ten constants in the order listed in Table 10 with their descriptions and units. Following the constants is a line with the number two which indicates that two independent design parameters will follow. These two parameters are the wire diameter and the mean coil diameter. Low, high and initial estimates are selected as 1.0, 15.0 and 5.0 for the wire size and 25.0, 500.0 and 100.0 for the mean coil diameter. The next line has the single value of seven for the seven design constraints listed in Table 10. All seven constraints happen to be lower. Following the constraints is a line with the letters 'MIN' to select minimization as the direction of optimization, a line with the value of three to indicate that there are three terms in the objective function. The last nine lines are the weighting coefficients, names and units for the three objective function terms of weight, height and volume. The weighting coefficient of the first is one and the last two are zero. By changing which term has the unit coefficient, one can change the optimization goal without changing the program.

Table 13

Spring Design Input File For Minimum Weight

HELICAL COIL SPRING - MINIMUM WEIGHT
10
24.0
MINIMUM FORCE
NEWTONS
120.0
MAXIMUM FORCE
NEWTONS
150.0
MINIMUM FORCE WHEN SOLID
NEWTONS
4.0
SPRING RATE
kN/m
2
END COIL NO.

465.0
SHEAR FATIGUE STRENGTH
MPa
1510.0
TENSILE STRENGTH CONSTANT
MPa
-0.201
TENSILE STRENGTH WIRE POWER

79000.0
SHEAR MODULUS
MPa
76.5
WEIGHT DENSITY
kN/m**3
2
1.0 15.0 5.0
WIRE DIAMETER
mm
25.0 500.0 100.0
MEAN COIL DIAMETER
mm

Table 13 Continued

Spring Design Input File For Minimum Weight

7
LOWER 0.0
ACTIVE COIL
NO.
LOWER 1.5
FATIGUE DESIGN FACTOR

LOWER 1.5
STATIC DESIGN FACTOR

LOWER 2.0
SPRING INDEX

LOWER 0.0
MAXIMUM SHEAR STRESS
MPa
LOWER 0.0
OUTSIDE COIL DIAMETER
mm
LOWER 0.0
SPRING WIRE VOLUME
mm**3
MIN
3
1.0
WEIGHT
NEWTONS
0.0
HEIGHT
mm
0.0
SPRING COIL VOLUME
mm**3

This ASCII file can then be saved with a name such as 'WEIGHT.IN' and used with the spring optimization program. Table 14 is the output file which resulted from running the spring optimization program with this file. The ten constants, two design parameters, seven constraints and three objective function term are all listed with their values, names, units, limit directions, and weighting coefficients at the start of the file. The optimization reached a solution in 24 steps with a wire diameter of slightly more than 3 mm and a mean coil diameter of 23.13 mm. The spring weight was 0.79 Newtons and the spring had 17.5 active coils with a height of 97 mm a spring index of 7.5 and a spring coil volume of 52,000 mm^3. The static overload stress was the limiting factor in the design with a static design factor of 1.5 at the limit and a maximum shear stress of 370 MPa.

Following this output is a design check with a 3.5 mm wire diameter and a mean coil diameter of 35 mm. With the larger standard wire size, the larger coil diameter keeps the spring weight down by reducing the number of active coils needed to obtain the spring rate without lowering the design factors. The weight increased to 0.86 Newtons, the number of active coils dropped to 8.64 and the spring index increased to 10. In addition, the height dropped to 75 mm and the spring coil volume increased to 87,500 mm^3. Although the spring is slightly heavier than the initial optimum, it has standard wire and coil dimensions, so it is a practical optimal solution to the posed problem.

Table 15 is the output file produced by running the spring optimization program with a second input file which has two weighting coefficients changed and the problem title changed in line one. The two numerical changes were to the first and second weighting coefficient values to switch the object of minimization from weight to height. As shown in Table 15, this optimum is

58

Table 14

Spring Design Output File For Minimum Weight

HELICAL COIL SPRING - MINIMUM WEIGHT

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1	MINIMUM FORCE	24.00000	NEWTONS
2	MAXIMUM FORCE	120.00000	NEWTONS
3	MINIMUM FORCE WHEN SOLID	150.00000	NEWTONS
4	SPRING RATE	4.00000	kN/m
5	END COIL NO.	2.00000	
6	SHEAR FATIGUE STRENGTH	465.00000	MPa
7	TENSILE STRENGTH CONSTANT	1510.00000	MPa
8	TENSILE STRENGTH WIRE POWER	-0.20100	
9	SHEAR MODULUS	79000.00000	MPa
10	WEIGHT DENSITY	76.50000	kN/m**3

THERE ARE 2 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

	LOW	HIGH	INITIAL	
1 WIRE DIAMETER	1.00000	15.00000	5.00000	mm
2 MEAN COIL DIAMETER	25.00000	500.00000	100.00000	mm

THE 7 CONSTRAINT LIMITS ARE:

				TYPE
1	ACTIVE COIL	0.00000	NO.	LOWER
2	FATIGUE DESIGN FACTOR	1.50000		LOWER
3	STATIC DESIGN FACTOR	1.50000		LOWER
4	SPRING INDEX	2.00000		LOWER
5	MAXIMUM SHEAR STRESS	0.00000	MPa	LOWER
6	OUTSIDE COIL DIAMETER	0.00000	mm	LOWER
7	SPRING WIRE VOLUME	0.00000	mm**3	LOWER

MINIMIZE THE OBJECTIVE FUNCTION, WHICH HAS 3 TERMS.

OBJ = THE LINEAR SUM OF:

1	WEIGHT	IN	NEWTONS	TIMES	1.0000
2	HEIGHT	IN	mm	TIMES	0.0000
3	SPRING COIL VOLUME	IN	mm**3	TIMES	0.0000

Table 14 Continued

Spring Design Output File For Minimum Weight

OPTIMIZATION SUCCESSFUL IN   24 STEPS

THE FINAL DESIGN VECTOR IS:

X(I)

1	WIRE DIAMETER	3.05491	mm
2	MEAN COIL DIAMETER	23.12969	mm

THE MINIMUM OBJECTIVE FUNCTION = 0.789480   , ITS COMPONENTS ARE:

1	WEIGHT	=0.78948	NEWTONS	TIMES	1.0000
2	HEIGHT	= 97.285	mm	TIMES	0.0000
3	SPRING COIL VOLUME	= 52388.	mm**3	TIMES	0.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION   = 0.596046E-07
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE = 0.390625E-03

THE 7 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 ACTIVE COIL	= 17.525	NO.	.00000	LOWER
2 FATIGUE DESIGN FACTOR	= 2.1345		1.5000	LOWER
3 STATIC DESIGN FACTOR	= 1.5035		1.5000	LOWER
4 SPRING INDEX	= 7.5462		2.0000	LOWER
5 MAXIMUM SHEAR STRESS	= 370.49	MPa	.00000	LOWER
6 OUTSIDE COIL DIAMETER	= 26.070	mm	.00000	LOWER
7 SPRING WIRE VOLUME	= 10320.	mm**3	.00000	LOWER

Table 14 Continued

Spring Design Output File For Minimum Weight

DESIGN CHECK

X(I)

1	WIRE DIAMETER	3.50000	mm
2	MEAN COIL DIAMETER	35.00000	mm

THE MINIMUM OBJECTIVE FUNCTION = 0.861137   ,   ITS COMPONENTS ARE:

1	WEIGHT	=0.86114	NEWTONS	TIMES	1.0000
2	HEIGHT	= 75.115	mm	TIMES	0.0000
3	SPRING COIL VOLUME	= 87445.	mm**3	TIMES	0.0000

THE   7 CONSTRAINT VALUES ARE:

				LIMIT	TYPE
1	ACTIVE COIL	= 8.6406	NO.	.00000	LOWER
2	FATIGUE DESIGN FACTOR	= 2.1430		1.5000	LOWER
3	STATIC DESIGN FACTOR	= 1.5179		1.5000	LOWER
4	SPRING INDEX	= 10.000		2.0000	LOWER
5	MAXIMUM SHEAR STRESS	= 356.97	MPa	.00000	LOWER
6	OUTSIDE COIL DIAMETER	= 38.500	mm	.00000	LOWER
7	SPRING WIRE VOLUME	= 11257.	mm**3	.00000	LOWER

Table 15

Spring Design Output File For Minimum Height

HELICAL COIL SPRING - MINIMUM HEIGHT

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1	MINIMUM FORCE	24.00000	NEWTONS
2	MAXIMUM FORCE	120.00000	NEWTONS
3	MINIMUM FORCE WHEN SOLID	150.00000	NEWTONS
4	SPRING RATE	4.00000	kN/m
5	END COIL NO.	2.00000	
6	SHEAR FATIGUE STRENGTH	465.00000	MPa
7	TENSILE STRENGTH CONSTANT	1510.00000	MPa
8	TENSILE STRENGTH WIRE POWER	-0.20100	
9	SHEAR MODULUS	79000.00000	MPa
10	WEIGHT DENSITY	76.50000	kN/m**3

THERE ARE 2 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

	LOW	HIGH	INITIAL	
1 WIRE DIAMETER	1.00000	15.00000	5.00000	mm
2 MEAN COIL DIAMETER	25.00000	500.00000	100.00000	mm

THE 7 CONSTRAINT LIMITS ARE:

				TYPE
1	ACTIVE COIL	0.00000	NO.	LOWER
2	FATIGUE DESIGN FACTOR	1.50000		LOWER
3	STATIC DESIGN FACTOR	1.50000		LOWER
4	SPRING INDEX	2.00000		LOWER
5	MAXIMUM SHEAR STRESS	0.00000	MPa	LOWER
6	OUTSIDE COIL DIAMETER	0.00000	mm	LOWER
7	SPRING WIRE VOLUME	0.00000	mm**3	LOWER

MINIMIZE THE OBJECTIVE FUNCTION, WHICH HAS 3 TERMS.

OBJ = THE LINEAR SUM OF:

1	WEIGHT	IN	NEWTONS	TIMES	0.0000
2	HEIGHT	IN	mm	TIMES	1.0000
3	SPRING COIL VOLUME	IN	mm**3	TIMES	0.0000

Table 15 Continued

Spring Design Output File For Minimum Height

OPTIMIZATION SUCCESSFUL IN   14 STEPS

THE FINAL DESIGN VECTOR IS:

X(I)

1	WIRE DIAMETER	6.30577	mm
2	MEAN COIL DIAMETER	195.96942	mm

THE MINIMUM OBJECTIVE FUNCTION =   53.5408   ,   ITS COMPONENTS ARE:

1	WEIGHT	= 3.7045	NEWTONS	TIMES	0.0000
2	HEIGHT	= 53.541	mm	TIMES	1.0000
3	SPRING COIL VOLUME	=0.17205E+07	mm**3	TIMES	0.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION   =   0.381470E-04
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE =   0.625000E-02

THE 7 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 ACTIVE COIL	= .53093	NO.	.00000	LOWER
2 FATIGUE DESIGN FACTOR	= 2.2086		1.5000	LOWER
3 STATIC DESIGN FACTOR	= 1.5438		1.5000	LOWER
4 SPRING INDEX	= 30.786		2.0000	LOWER
5 MAXIMUM SHEAR STRESS	= 312.12	MPa	.00000	LOWER
6 OUTSIDE COIL DIAMETER	= 199.46	mm	.00000	LOWER
7 SPRING WIRE VOLUME	= 47506.	mm**3	.00000	LOWER

Table 15 Continued

Spring Design Output File For Minimum Height

DESIGN CHECK

X(I)

1	WIRE DIAMETER	6.50000	mm
2	MEAN COIL DIAMETER	220.00000	mm

THE MINIMUM OBJECTIVE FUNCTION =   53.3470   ,   ITS COMPONENTS ARE:

1	WEIGHT	= 4.2351	NEWTONS	TIMES	0.0000
2	HEIGHT	= 53.347	mm	TIMES	1.0000
3	SPRING COIL VOLUME	=0.21495E+07	mm**3	TIMES	0.0000

THE   7 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 ACTIVE COIL	= .41387	NO.	.00000	LOWER
2 FATIGUE DESIGN FACTOR	= 2.1515		1.5000	LOWER
3 STATIC DESIGN FACTOR	= 1.5020		1.5000	LOWER
4 SPRING INDEX	= 33.846		2.0000	LOWER
5 MAXIMUM SHEAR STRESS	= 318.54	MPa	.00000	LOWER
6 OUTSIDE COIL DIAMETER	= 226.50	mm	.00000	LOWER
7 SPRING WIRE VOLUME	= 55361.	mm**3	.00000	LOWER

different from the first. The spring has a wire diameter of 6.3 mm and a mean coil diameter of 196 mm with only 0.53 active coils and a spring index of 31. The height is shorter at 53.5 mm but the weight is higher at 3.7 Newtons and the spring coil volume is much higher at $1.72*10^6$ mm^3. This design was also limited by the static design factor.

However, it was reached in only 14 trials and the static design factor was slightly higher than 1.5 at 1.54. These two facts indicate that this may not be an absolute minimum height design. Changing the starting position as was done for the bushing design or changing the sensitivity ranges on the two independent design parameters would give the optimizer a chance to search longer and find a slightly better optimal design. However, since we were going to change the wire size and mean coil diameter to standard values anyway, the starting position and sensitivity ranges were not changed. The design check chosen has a wire diameter of 6.5 mm and a mean coil diameter of 220 mm for a height of 53.3 mm. This design is also heavy and large with a fraction of an active coil at 0.41 and a spring index of 34.

Table 16 shows the output file for a design which minimizes the spring coil volume. The only changes in the input file for this case were the weighting coefficients in the objective function list to select spring volume as the target for minimization and the problem title. This design was achieved in 28 steps and has a wire diameter of 2.2 mm and a mean coil diameter of 7.5 mm. The spring coil volume is the smallest of the found designs at 26,300 mm^3 which is about one-half that of the first design. However it is heavier at 0.99 Newtons and much longer at 355 mm. The spring index is small at 3.4 and the number of active coils is large at 140. Changing the wire diameter to 2.5 mm and the mean coil diameter to 11.5 mm

65

Table 16

Spring Design Output File For Minimum Coil Volume

HELICAL COIL SPRING - MINIMUM COIL VOLUME

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1	MINIMUM FORCE	24.00000	NEWTONS
2	MAXIMUM FORCE	120.00000	NEWTONS
3	MINIMUM FORCE WHEN SOLID	150.00000	NEWTONS
4	SPRING RATE	4.00000	kN/m
5	END COIL NO.	2.00000	
6	SHEAR FATIGUE STRENGTH	465.00000	MPa
7	TENSILE STRENGTH CONSTANT	1510.00000	MPa
8	TENSILE STRENGTH WIRE POWER	-0.20100	
9	SHEAR MODULUS	79000.00000	MPa
10	WEIGHT DENSITY	76.50000	kN/m**3

THERE ARE 2 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

	LOW	HIGH	INITIAL	
1 WIRE DIAMETER	1.00000	15.00000	5.00000	mm
2 MEAN COIL DIAMETER	25.00000	500.00000	100.00000	mm

THE 7 CONSTRAINT LIMITS ARE:

				TYPE
1	ACTIVE COIL	0.00000	NO.	LOWER
2	FATIGUE DESIGN FACTOR	1.50000		LOWER
3	STATIC DESIGN FACTOR	1.50000		LOWER
4	SPRING INDEX	2.00000		LOWER
5	MAXIMUM SHEAR STRESS	0.00000	MPa	LOWER
6	OUTSIDE COIL DIAMETER	0.00000	mm	LOWER
7	SPRING WIRE VOLUME	0.00000	mm**3	LOWER

MINIMIZE THE OBJECTIVE FUNCTION, WHICH HAS 3 TERMS.

OBJ = THE LINEAR SUM OF:

1	WEIGHT	IN	NEWTONS	TIMES	0.0000
2	HEIGHT	IN	mm	TIMES	0.0000
3	SPRING COIL VOLUME	IN	mm**3	TIMES	1.0000

66

Table 16 Continued

Spring Design Output File For Minimum Coil Volume

OPTIMIZATION SUCCESSFUL IN   28 STEPS

THE FINAL DESIGN VECTOR IS:

                                        X(I)

1	WIRE DIAMETER	2.21384	mm
2	MEAN COIL DIAMETER	7.51284	mm

THE MINIMUM OBJECTIVE FUNCTION =   26353.6   ,   ITS COMPONENTS ARE:

1	WEIGHT	=0.98587	NEWTONS	TIMES	0.0000
2	HEIGHT	= 354.67	mm	TIMES	0.0000
3	SPRING COIL VOLUME	= 26354.	mm**3	TIMES	1.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION   = -0.781250E-02
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE = 0.195313E-03

THE   7 CONSTRAINT VALUES ARE:

				LIMIT	TYPE
1	ACTIVE COIL	= 140.58	NO.	.00000	LOWER
2	FATIGUE DESIGN FACTOR	= 2.2583		1.5000	LOWER
3	STATIC DESIGN FACTOR	= 1.5032		1.5000	LOWER
4	SPRING INDEX	= 3.3870		2.0000	LOWER
5	MAXIMUM SHEAR STRESS	= 395.29	MPa	.00000	LOWER
6	OUTSIDE COIL DIAMETER	= 9.7067	mm	.00000	LOWER
7	SPRING WIRE VOLUME	= 12907.	mm**3	.00000	LOWER

Table 16 Continued

Spring Design Output File For Minimum Coil Volume

DESIGN CHECK

X(I)

1	WIRE DIAMETER	2.50000	mm
2	MEAN COIL DIAMETER	11.50000	mm

THE MINIMUM OBJECTIVE FUNCTION =   31196.3   ,   ITS COMPONENTS ARE:

1	WEIGHT	=0.88738	NEWTONS	TIMES	0.0000
2	HEIGHT	= 202.66	mm	TIMES	0.0000
3	SPRING COIL VOLUME	= 31196.	mm**3	TIMES	1.0000

THE   7 CONSTRAINT VALUES ARE:

				LIMIT	TYPE
1	ACTIVE COIL	= 63.408	NO.	.00000	LOWER
2	FATIGUE DESIGN FACTOR	= 2.2430		1.5000	LOWER
3	STATIC DESIGN FACTOR	= 1.5367		1.5000	LOWER
4	SPRING INDEX	= 4.6000		2.0000	LOWER
5	MAXIMUM SHEAR STRESS	= 377.29	MPa	.00000	LOWER
6	OUTSIDE COIL DIAMETER	= 14.000	mm	.00000	LOWER
7	SPRING WIRE VOLUME	= 11600.	mm**3	.00000	LOWER

with the design check provision gives a design with a spring coil volume of 31,200 mm^3, a weight of only 0.89 Newtons and a height of 203 mm. This spring has 63.4 active coils with a spring index of 4.6.

Figure 11 shows all three optimum designs to the same scale for comparison. In Figure 10, which was used earlier to describe the design space, three crosses mark the locations of the three optimal designs in this design space. Each design is at or near the static strength limit, and the minimum weight design, with its cross between the other two, is the design we expect from practice. It is the most economical in that it uses the least material and it is not extremely long or large in diameter. However, the other designs may still have their applications.
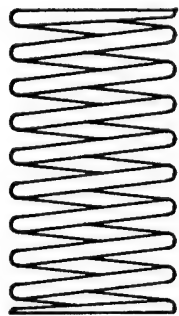
MINIMUM WEIGHT   MINIMUM VOLUME
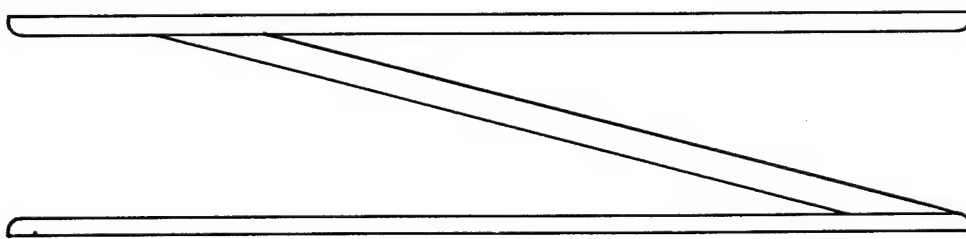
MINIMUM HEIGHT

OPTIMUM SPRING DESIGNS
FIGURE 11

SPUR GEAR OPTIMIZATION

The third example problem is that of a compact gear mesh, which is to have a compact size at a given life, load, reduction and speed. For steel involute teeth, the loading which may cause failure is dynamic due to the variation in contact geometry and load sharing as the teeth enter and leave the gear mesh. Three primary modes of failure are possible: 1) bending tooth fracture, 2) tooth surface pitting, and 3) tooth tip scoring. When fracture of the gear teeth due to bending is the primary mode of failure, the minimum number of teeth which avoids interference offers the strongest gear set for a given size [16]. However, as speeds increase, so do the prospects for pitting and scoring modes of failure.

For a given combination of gear material and lubrication conditions, the design problem can be formulated in terms of three independent design variables: The number of pinion teeth, N_p; the diametral pitch, P_d; and the gear and pinion face width, f. Although many designs can transmit the same power at the same input and output speeds, two designs will be sought. The first will have the minimum center distance between the input and output shafts for a specified reliability life. This design will also have the minimum gear weight and volume. The second will have the maximum life for a specified center distance.

Theory

All acceptable gear designs avoid involute interference. For equal pinion and gear addenda, involute interference will occur at the base of the pinion tooth. As shown in Figure 12, involute interference can be measured by the roll angle, δ_1, which corresponds to the distance along the line of action from the base circle of the pinion to the addendum circle of the gear.

71
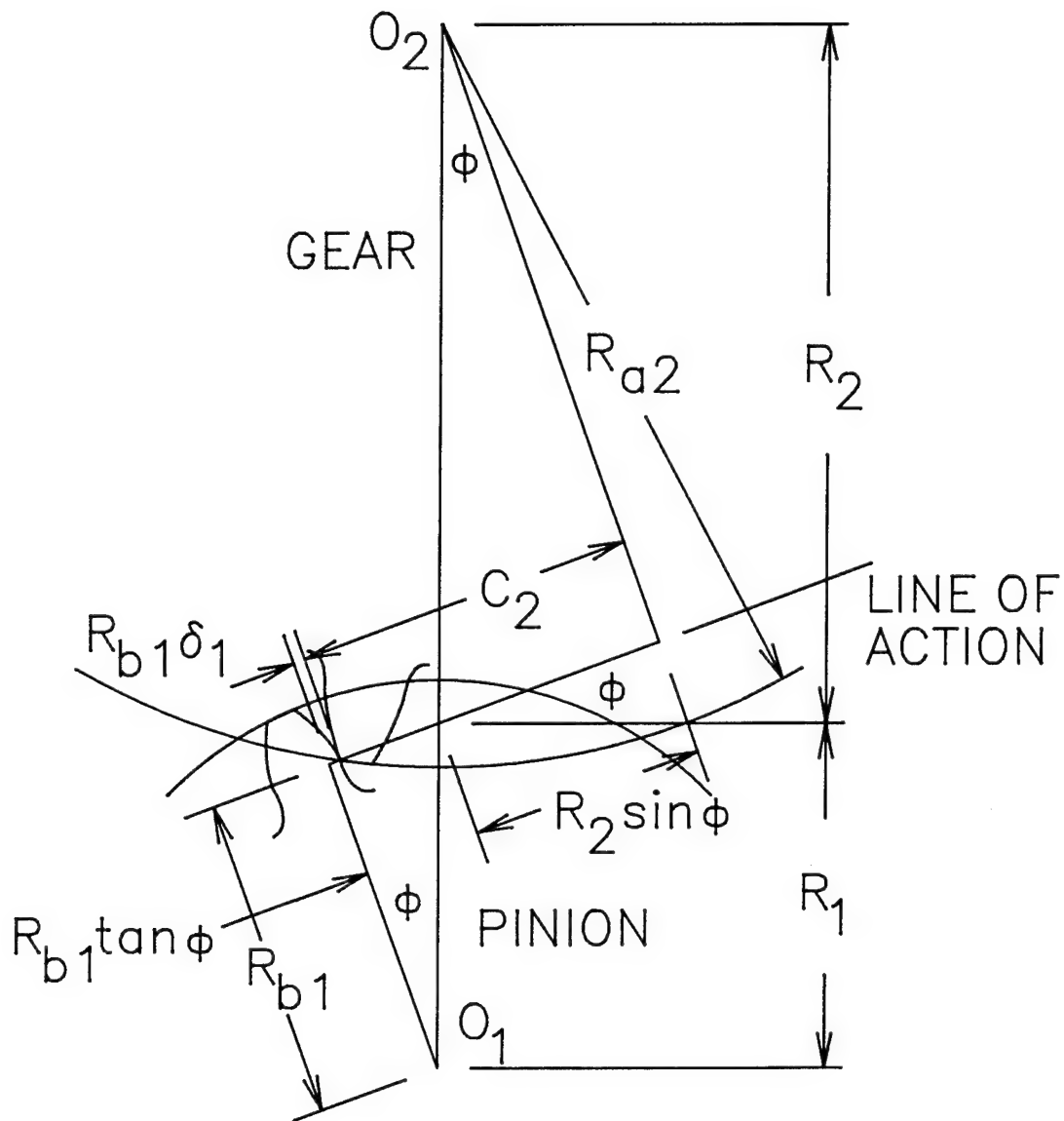
INVOLUTE INTERFERENCE CHECK

FIGURE 12

$$\delta_1 = \tan \phi + \frac{R_2 \sin \phi - \sqrt{R_{a2}^2 - R_{b2}^2}}{R_{b1}} \qquad (26)$$

This angle must be positive for the gear tooth tip to contact the pinion tooth on its involute surface and avoid interference.

In performing the gear tooth load analysis, the tangential load on the gear mesh, F_t, is the pinion torque divided by the pitch radius of the pinion. This is the nominal force acting between the gears. The force along the line of action is this force divided by the cosine of the pressure angle, ϕ. The pitch line velocity is the rotational speed of the pinion times its pitch radius, which is the speed of the pitch circles.

To estimate the dynamic load, one can use the AGMA velocity factor model [17]. In terms of a gear quality number, Q_v, the AGMA estimate of the sum of the transmitted load and the dynamic load is:

$$F_d = F_t \left(\frac{A + \sqrt{V}}{A} \right) \qquad (27)$$

where

$$A = 50 + 56 \left(1 - \frac{[12 - Q_v]^{2/3}}{4} \right) \qquad (28)$$

In equation (28), the gear quality number, Q_v, may have a value between 6 and 11 with 11 corresponding to the higher quality gear. In this example, all gear stresses and lives are calculated using this total dynamic load, F_d, with a quality number, $Q_v = 9$.

As noted above, gear tooth bending fatigue, gear tooth pitting and gear tip scoring are the three most probable modes of failure for gear teeth. The bending fatigue model uses the AGMA J factor [17] to estimate the bending

73

stress with the dynamic load at the highest point of single tooth contact on the pinion, which produces the maximum tooth bending stress. The formula for the bending stress is:

$$\sigma_b = \frac{F_d \cdot P_d}{f \cdot J} \tag{29}$$

Figure 13 shows the dynamic load on the gear tooth and the Lewis parabola which describes the strongest constant strength beam inscribed in the tooth.

Gear tooth pitting is a result of contact stress on the gear tooth surface. The maximum contact stress and gear tip Hertzian pressure are calculated [18] as:

$$\sigma_H = \left(\frac{F_d}{\pi f \cos \phi} \left(\frac{1 / \rho_1 + 1 / \rho_2}{\frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2}} \right) \right)^{1/2} \tag{30}$$

This maximum contact stress occurs at the lowest point of full load contact on the pinion tooth. Figure 14 shows the two teeth in contact at this point and their radii of curvature on which this stress is based. The small parabola on the tooth surface shows this contact pressure distribution.

The gear tip Hertzian pressure uses one-third of the total dynamic load since the load is shared unequally between two tooth pairs at this point due to the elastic interaction of the two tooth pairs in contact. The gear tip scoring model includes the pressure times velocity factor and the critical oil scoring temperature model from lubrication theory. The normal pressure times sliding velocity is proportional to the frictional power loss of the gear set. This factor is the highest for contact at the gear tip, with the normal

74
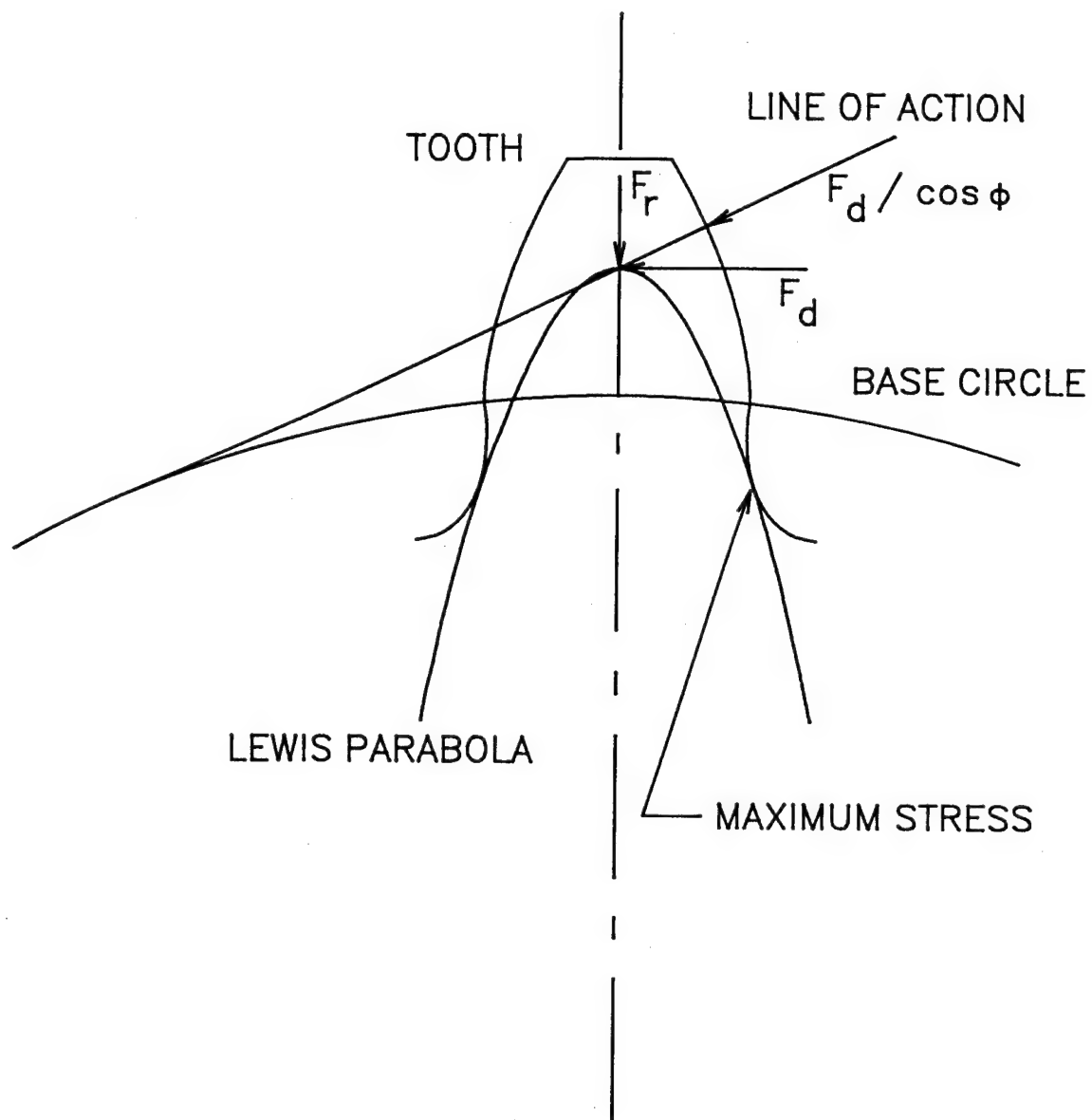
GEAR TOOTH BENDING STRENGTH MODEL

FIGURE 13
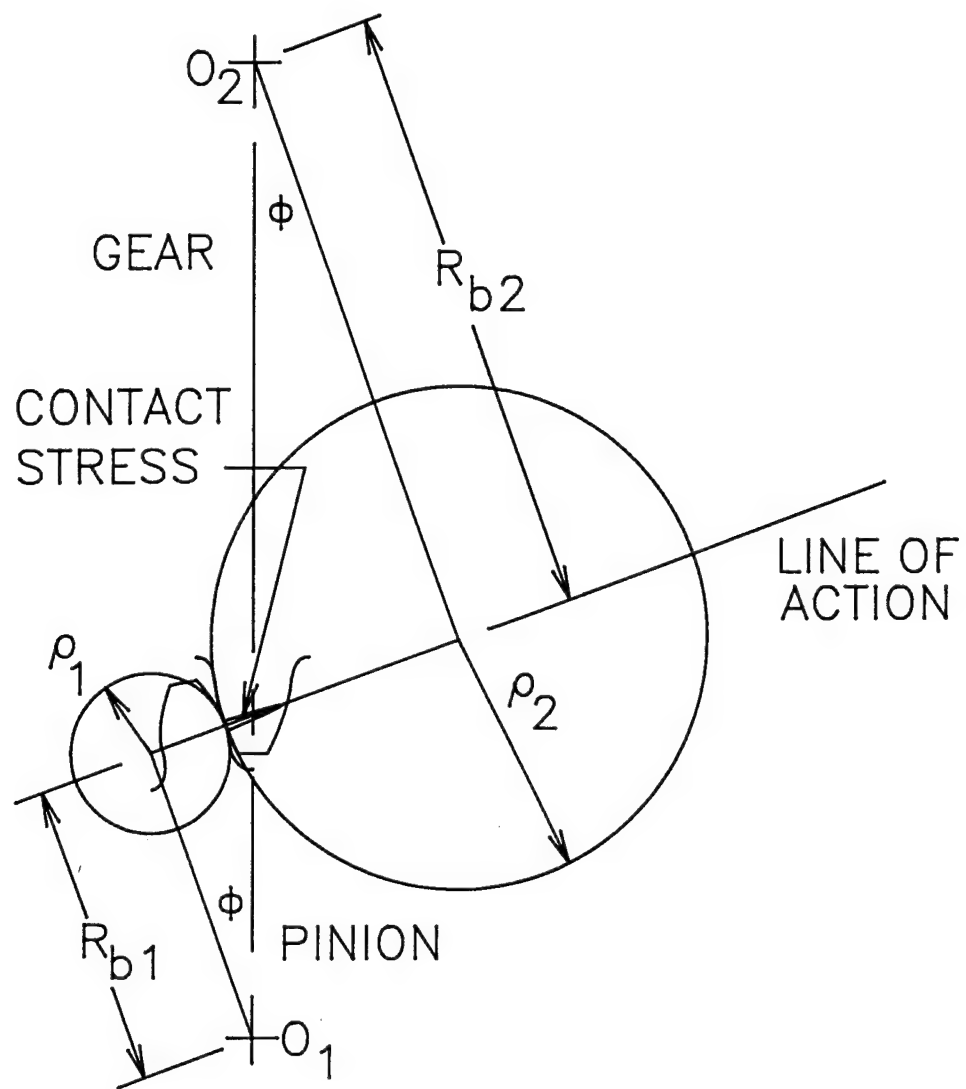
PINION AND GEAR
TOOTH SURFACE CURVATURE

FIGURE 14

pressure equal to the gear tip Hertzian pressure. The sliding velocity at the gear tip is given by:

$$V_s = \omega_2 R_{a2} \sin(\phi + \alpha_{a2}) - \omega_1 R_1 \sin(\phi + \alpha_1) \qquad (31)$$

The lubricating oil flash temperature is another factor used to monitor gear tooth scoring. One estimate of this temperature [18] is given by:

$$T_f = T_B + \left(\frac{X_\Gamma F_d}{f} \right)^{3/4} \left(\frac{0.45 \, \mu_m \, X_M \, X_G \sqrt{V}}{(R_1 + R_2)^{1/4}} \right) \qquad (32)$$

The gear life model is based on surface pitting and is similar to the model for rolling element bearings [19]. Its reliability estimation is based on the two-parameter Weibull distribution:

$$Ln \left(\frac{1}{R} \right) = Ln \left(\frac{1}{0.9} \right) \cdot \left(\frac{\ell}{\ell_{10}} \right)^b \qquad (33)$$

The life to reliability relationship of equation (33) is for a specific load which determines the ℓ_{10} life. This load, F, is related to the component dynamic capacity, C, as:

$$\ell_{10} = \left(\frac{C}{F} \right)^p \qquad (34)$$

where the dynamic capacity of the component, C, is the load which has a 90 percent reliability life of one million cycles.

For a spur gear tooth, the dynamic capacity, C_t, can be expressed as a function of Buckingham's load-stress factor [20], B, which is a material strength constant:

77

$$c_t = B \left(\frac{f}{1/\rho_1 + 1/\rho_2} \right) \tag{35}$$

With the dynamic capacity expressed in this form, the material strength constant serves the role of the surface fatigue strength, S_{ac}, of the AGMA design code. A relation for the material strength constant in terms of the surface fatigue strength is:

$$B = \pi S_{ac}^2 \left(\frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2} \right) \tag{36}$$

The dynamic capacity of the whole gear is lower than that of a single tooth due to the number of teeth subjected to the same load:

$$c_g = \frac{c_t}{N_g^{1/(b \cdot p)}} \tag{37}$$

The gear and pinion weights are modeled with solid discs which have radii equal to the pitch radii of the two gears and thicknesses equal to the face width.

Programming

Table 17 summarizes this design problem in terms of the constants, design parameters, inequality constraints and objective functions. There are thirteen constants, three design parameters, fourteen constraints and two objective functions in this list.

Included in the list of constants are: Poisson's ratio and the elastic modulus for the gear material: the nominal pressure angle, ϕ, and gear ratio, n, for the mesh; the transmitted power, Hp, and pinion speed, ω_1; the material weight density, γ, surface constant, B, Weibull slope, b, and load-life

Table 17

Spur Gear Mesh Optimization Parameters

Constants	Design Parameters	Inequality Constraints	Objective Function
v	N_1	$\delta_1 < 0.001$	C_{min}
E	P_d	$f/D_1 < 0.5$	or
ϕ	f	$C > 0.0$	$(\ell_m)_{max}$
n		$W_1 + W_2 > 0.0$	
Hp		$T_1 > 0.0$	
ω_1		$F_t > 0.0$	
γ		$V > 0.0$	
B		$F_d > 0.0$	
b		$\sigma_b < 40,000.$	
p		$\sigma_H < 150,000.$	
R		$\sigma_{Ht} < 150,000.$	
T_B		$P\,V_s < 100.$	
μ_m		$T_f < 275.$	
		$\ell_m > 2.0$	

exponent, p; the desired reliability of the mesh, R; the lubricant base temperature, T_B; and the surface finish, μ_m.

The three design parameters are: 1) the number of teeth on the pinion, N_1; the diametral pitch, P_d; and the face width, f.

Design constraints listed are: active constraints for involute interference and the width to diameter ratio; inactive constraints of being positive for the center distance, weight sum, input torque, transmitted force, pitch line velocity, and dynamic force; active bending and pitting strength limits; active scoring limits of pressure times velocity and flash temperature; and the mesh life bound.

The objective functions include the center distance which is to be minimized and the mesh life which is to be maximized. In running the program, these two functions will not be active in the same case. The listed values are for the first objective function of minimum center distance. When the program is run to maximize the mesh life, the center distance limit is changed to an active limit of 2.5 inches and the mesh life constraint is changed to be positive and thus inactive.

Subroutines BOUNDS and VALUES, developed for this example gear problem, are listed together in Table 18. These two routines call a series of analysis subroutines to evaluate the properties of the gear design. These subroutines: DYNAM, MESH, GLIFE, GEARWT and TEMPER are also listed in Table 18 along with LEWIS which MESH calls. Subroutine DYNAM determines the dynamic load in the mesh with the AGMA velocity factor calculation using equations (27) and (28). Subroutine MESH performs a kinematic analysis of the gear mesh geometry and calculates the bending stresses and Hertzian contact stresses on the teeth in the mesh with equations (26) and (29) through (31). MESH also calculates the

80

Table 18

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
      SUBROUTINE BOUNDS(CONST,NCO,X,NX,VCSTR,NCS)
C
C                    BOUNDS DETERMINES THE PRESENT CONSTRAINT
C                              FUNCTION VALUES
C
C             FOR A SPUR GEAR LIFE DESIGN EXAMPLE
C
C     PARAMETERS:
C
C             CONST - FIXED DESIGN CONSTANT
C             NCO   - NUMBER OF DESIGN CONSTANTS
C             NCS   - NUMBER OF INEQUALITY CONSTRAINTS
C             NX    - NUMBER OF INDEPENDENT DESIGN VARIABLES
C             VCSTR - PRESENT CONSTRAINT VALUES
C             X     - PRESENT DESIGN VARIABLE VALUES
C
C             ALL VALUES ARE IN PROBLEM UNITS (UNSCALED)
C
C             CONST(1) = mu    - POISSON'S RATIO
C             CONST(2) = E     - ELASTIC MODULUS (PSI)
C             CONST(3) = PHI   - PRESSURE ANGLE (DEGREES)
C             CONST(4) = N     - GEAR RATIO
C             CONST(5) = Hp    - PINION POWER (HORSEPOWER)
C             CONST(6) = Np    - PINION SPEED (RPM)
C             CONST(7) = GAMMA - MATERIAL WEIGHT DENSITY (LBS/IN**3)
C             CONST(8) = Bc    - MATERIAL STRENGTH CONSTANT (PSI)
C             CONST(9) = b     - WEIBULL SLOPE
C             CONST(10)= p     - LOAD-LIFE FACTOR
C             CONST(11)= Rel   - DESIGN RELIABILITY
C             CONST(12)= Tb    - OIL INLET TEMPERATURE (DEGREES F)
C             CONST(13)= RMS   - TOOTH SURFACE FINISH (RMS)
C
C             VCSTR(1) = DEL1   - INVOLUTE INTERFERENCE ANGLE (RADIANS)
C             VCSTR(2) = LAMMIN - FACE WIDTH TO PINION DIAMETER RATIO
C             VCSTR(3) = C      - CENTER DISTANCE (INCHES)
C             VCSTR(4) = WT     - PINION & GEAR WEIGHTS (POUNDS)
C             VCSTR(5) = Tq     - PINION TORQUE (POUND - INCHES)
C             VCSTR(6) = Ft     - TRANSMITTED FORCE (POUNDS)
C             VCSTR(7) = V      - PITCH LINE VELOCITY (FT/MIN)
C             VCSTR(8) = Fd     - DYNAMIC LOAD (POUNDS)
C             VCSTR(9) = BDSTR  - AGMA BENDING STRESS (PSI)
C             VCSTR(10)= HZSTR  - CONTACT PRESSURE (PSI)
C             VCSTR(11)= TIPPR  - GEAR TIP HERTZ CONTACT PRESSURE (PSI)
C             VCSTR(12)= PVF    - PRESSURE TIMES VELOCITY FACTOR
C                                         (10**6 PSI-FT/MIN)
C             VCSTR(13)= Tf     - FLASH TEMPERATURE (DEGREES F)
C             VCSTR(14)= Lm     - MESH LIFE (10**3 HOURS)
C
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
C             X(1) = n1        - NUMBER OF PINION TEETH
C             X(2) = Pd        - DIAMETRAL PITCH (1.0/IN)
C             X(3) = Wd        -- FACE WIDTH (INCHES)
C
C
      DIMENSION X(NX),VCSTR(NCS),CONST(NCO)
      REAL LM,LP,LG
      PI = 3.14159265
      POI = CONST(1)
      E = CONST(2)
      PHI = CONST(3)
      RATIO = CONST(4)
      HP = CONST(5)
      RPM = CONST(6)
      GAM = CONST(7)
      BC = CONST(8)
      B = CONST(9)
      P = CONST(10)
      REL = CONST(11)
      TB = CONST(12)
      RMS = CONST(13)
      TN1 = X(1)
      PD = X(2)
      WD = X(3)
      TN2 = RATIO*TN1
      PHIR = PHI*PI/180.0
      RP = 0.5*TN1/PD
      RG = RATIO*RP
      C = RP*( 1.0 + RATIO )
      V = RPM*RP*PI/6.0
      TQ = HP*63025.0/RPM
      FT = TQ/RP
      CALL DYNAM(FT,V,DL)
      TDQ = RP*DL
      CALL MESH(PHI,PD,TN1,TN2,E,POI,WD,TDQ,RPM,CMP,DEL1,RH1,
     1  PVF,TIPHZ,TIPBS,HZSTR,BDSTR)
      RH2 = C*SIN(PHIR) - RH1
      CALL GLIFE(DL,WD,RH1,RH2,TN1,TN2,RPM,BC,B,P,REL,
     1  LM,LP,LG,CYLP,CYLG)
      CALL GEARWT(GAM,GAM,RP,RG,WD,PW,GW)
      CALL TEMPER(PD,PHI,TN1,TN2,WD,DL,V,TB,RMS,TF,CSN)
      VCSTR(1) = DEL1
      VCSTR(2) = WD/(2.0*RP)
      VCSTR(3) = C
      VCSTR(4) = PW + GW
      VCSTR(5) = TQ
      VCSTR(6) = FT
      VCSTR(7) = V
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
      VCSTR(8) = DL
      VCSTR(9) = BDSTR
      VCSTR(10) = HZSTR
      VCSTR(11) = TIPHZ
      VCSTR(12) = PVF/1000000.0
      VCSTR(13) = TF
      VCSTR(14) = LM/1000.0
      RETURN
      END
C
      SUBROUTINE VALUES(CONST,NCO,X,NX,OBJECT,NOB)
C
C               VALUES DETERMINES THE PRESENT DESIGN
C                     OBJECTIVE FUNCTION VALUES
C
C
C         PARAMETERS:
C
C               CONST  - FIXED DESIGN CONSTANT
C               NCO    - NUMBER OF DESIGN CONSTANTS
C               NOB    - NUMBER OF OBJECTIVE FUNCTION TERMS
C               NX     - NUMBER OF INDEPENDENT DESIGN VARIABLES
C               OBJECT - PRESENT OBJECTIVE FUNCTION VALUES
C               X      - PRESENT DESIGN VARIABLE VALUES
C
C         ALL VALUES ARE IN PROBLEM UNITS (UNSCALED)
C
C               CONST(1) = mu    - POISSON'S RATIO
C               CONST(2) = E     - ELASTIC MODULUS (PSI)
C               CONST(3) = PHI   - PRESSURE ANGLE (DEGREES)
C               CONST(4) = N     - GEAR RATIO
C               CONST(5) = Hp    - PINION POWER (HORSEPOWER)
C               CONST(6) = Np    - PINION SPEED (RPM)
C               CONST(7) = GAMMA - MATERIAL WEIGHT DENSITY (LBS/IN**3)
C               CONST(8) = Bc    - MATERIAL STRENGTH CONSTANT (PSI)
C               CONST(9) = b     - WEIBULL SLOPE
C               CONST(10)= p     - LOAD-LIFE FACTOR
C               CONST(11)= Rel   - DESIGN RELIABILITY
C               CONST(12)= Tb    - BASE TEMPERATURE (DEG FAHR)
C               CONST(13)= RMS   - TOOTH SURFACE FINISH (RMS)
C
C                  OBJECT(1) = C          - CENTER DISTANCE (INCHES)
C                  OBJECT(2) = Lm         - MESH LIFE (10**3 HOURS)
C
C                  X(1) = nl          - NUMBER OF PINION TEETH
C                  X(2) = Pd          - DIAMETRAL PITCH (1.0/IN)
C                  X(3) = Wd          - FACE WIDTH (INCHES)
C
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
C
      DIMENSION X(NX),OBJECT(NOB),CONST(NCO)
      REAL LM,LP,LG
      PI = 3.14159265
      POI = CONST(1)
      E = CONST(2)
      PHI = CONST(3)
      RATIO = CONST(4)
      HP = CONST(5)
      RPM = CONST(6)
      BC = CONST(8)
      B = CONST(9)
      P = CONST(10)
      REL = CONST(11)
      TN1 = X(1)
      PD = X(2)
      WD = X(3)
      TN2 = RATIO*TN1
      PHIR = PHI*PI/180.0
      RP = 0.5*TN1/PD
      C = RP*( 1.0 + RATIO )
      V = RPM*RP*PI/6.0
      TQ = HP*63025.0/RPM
      FT = TQ/RP
      CALL DYNAM(FT,V,DL)
      TDQ = RP*DL
      CALL MESH(PHI,PD,TN1,TN2,E,POI,WD,TDQ,RPM,CMP,DEL1,RH1,
     1  PVF,TIPHZ,TIPBS,HZSTR,BDSTR)
      RH2 = C*SIN(PHIR) - RH1
      CALL GLIFE(DL,WD,RH1,RH2,TN1,TN2,RPM,BC,B,P,REL,
     1  LM,LP,LG,CYLP,CYLG)
      OBJECT(1) = C
      OBJECT(2) = LM/1000.0
      RETURN
      END
C
      SUBROUTINE MESH(PHI,PD,TN1,TN2,EL,POI,WD,TQ,RPM,CMP,DEL1,ROB,
     1  PVF,SQA,AGBN2,SQB,AGBN1)
C
C                         MESH
C
C
C     THIS ROUTINE CALCULATES THE GEAR TOOTH CONTACT GEOMETRY
C
C           INPUTS:
C                   PHI    - NOMINAL PRESSURE ANGLE (DEGREES)
C                   PD     - DIAMETRAL PITCH (1.0/INCH)
C                   TN1    - NUMBER OF TEETH ON PINION
C                   TN2    - NUMBER OF TEETH ON GEAR
```

84

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
C                      EL    - ELASTIC MODULUS OF TEETH (PSI)
C                      POI   - POISSON'S RATIO FOR TEETH
C                      WD    - TOOTH FACE WIDTH (INCHES)
C                      TQ    - TORQUE APPLIED TO PINION (POUND - INCHES)
C                      RPM   - SPEED OF PINION IN REVOLUTIONS PER MINUTE
C
C          OUTPUTS:
C                      CMP   - MESH CONTACT RATIO
C                      DEL1  - DISTANCE FROM PINION BASE CIRCLE TANGENCY
C                                 TO TIP OF GEAR TOOTH ALONG LINE OF ACTION
C                                 MEASURED AS A PINION ROLL ANGLE
C                      ROB   - PINION TOOTH SURFACE RADIUS OF CURVATURE
C                                 AT THE LOWEST POINT OF SINGLE TOOTH CONTACT C
         PVF      - SCORING FACTOR (PSI-FT/MIN / 10**6)
C                      SQA    - HERTZIAN CONTACT STRESS AT TIP OF GEAR TOOTH
C                                                                  (PSI)
C                      AGBN2 - AGMA BENDING STRESS FOR HALF LOAD AT TIP
C                                          OF GEAR TOOTH (PSI)
C                      SQB    - HERTZIAN CONTACT STRESS AT LOWEST POINT OF
C                                          SINGLE TOOTH CONTACT (PSI)
C                      AGBN1 - AGMA BENDING STRESS FOR FULL LOAD AT
C                                          HIGHEST POINT OF SINGLE TOOTH CONTACT (PSI) C
         ACS(A)=ATAN(SQRT(1.-A*A)/A)
         ASN(A)=ATAN(A/SQRT(1.-A*A))
         FINV(A)=SIN(A)/COS(A)-A
         NW=0
         PI=3.14159265
C
C        IT IS ASSUMED THAT THE PINION IS THE DRIVING GEAR
C
         J=1
         FJ=J
         ADD=1.
         DED=1.25
         RFR=0.3
         IF(PD.GE.20.0) DED=1.2+0.002*PD
         ADD1=ADD
         ADD2=ADD
         DDD1=DED
         AD1=ADD1/PD
         AD2=ADD2/PD
         DD1=DDD1/PD
         RF=RFR/PD
         PC=PI/PD
         RAD=180./PI
         P=PHI/RAD
         COP=COS(P)
         SIP=SIN(P)
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
PB=PC*COP
R1=0.5*PC*TN1/PI
R2=0.5*PC*TN2/PI
RB1=R1*COP
RB2=R2*COP
RA1=R1+AD1
RA2=R2+AD2*FJ
C=FJ*R1+R2
C1=SQRT(RA1*RA1-RB1*RB1)
C2=SQRT(RA2*RA2-RB2*RB2)
Z=C1+(C2-C*SIP)*FJ
CMP=Z/PB
AA1=(C2-R2*SIP)/RB1*FJ
AA2=(C2-R2*SIP)/RB2*FJ
DEL1=SIP/COP-AA1
C
C       IF DEL1 LESS THAN 0 INTERFERENCE WILL HAPPEN
C
C               SET EQUAL TO A SMALL VALUE TO GENERATE
C               LARGE CONTACT STRESSES
C
IF(DEL1.LT.0.0) DEL1 = 0.001*SIP/COP
AB2=(C1-R1*SIP)/RB2
DEL2=SIP/COP-AB2
IF(J.LT.0) DEL2=SIP/COP-AA2
C
C       IF DEL2 LESS THAN 0 INTERFERENCE WILL HAPPEN
C
C               SET EQUAL TO A SMALL VALUE TO GENERATE
C               LARGE CONTACT STRESSES
C
IF(DEL2.LT.0.0) DEL2 = 0.001*SIP/COP
BL1=(Z-PB)/RB1
BH1=(2.*PB-Z)/RB1
TL1=DEL1+BL1
TU1=TL1+BH1
RAT=TN2/TN1
IF(J.GT.0) GO TO 5
TP=FINV(P)
PA1=ACS(RB1/RA1)
PA2=ACS(RB2/RA2)
TA1=FINV(PA1)
TA2=FINV(PA2)
BETA1=ACS((RA2*RA2-RA1*RA1-C*C)/(2.0*C*RA1))
BETA2=ASN(RA1*SIN(BETA1)/RA2)
GAMMA1=BETA1+TA1-TP
GLIM2=BETA2+TA2-TP
GAMMA2=TN1*GAMMA1/TN2
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
      IF(GAMMA2.GT.GLIM2) GO TO 5
      XINT=RA2*(GLIM2-GAMMA2)
      WRITE (NW,122) XINT
  122 FORMAT(//'             SECONDARY INVOLUTE INTERFERENCE'/
     1 '        TOOTH TIP OVERLAP = ',F8.5,'  INCHES AT ENTRY'/
     2 '        INCREASE N2/N1 TO AVOID THIS INTERFERENCE'/)
    5 CONTINUE
      FSIN = TQ/R1
      FTIP = FSIN/3.0
      EO=EL/(1.-POI*POI)
      ROA=RB1*DEL1
      ROB=RB1*TL1
      CAP1=1./(ROA*(1.-FJ*ROA/(C*SIP)))
      CAP2=1./(ROB*(1.-FJ*ROB/(C*SIP)))
      SBA=SQRT(2.*FTIP/(COP*PI*EO*WD*CAP1))
      SQA=FTIP/(COP*PI*WD*SBA)
      SBB=SQRT(2.*FSIN/(COP*PI*EO*WD*CAP2))
      SQB=FSIN/(COP*PI*WD*SBB)
      PU1=ATAN(TU1)*RAD
      TC=PC/2.
      CALL LEWIS(PHI,DD1,TC,RF,PD,TN1,PU1,YS,AKS)
      SBND1=FSIN*PD/(WD*YS)
      AGBN1=SBND1*AKS
      TT1=TU1+BL1
      PT1=ATAN(TT1)*RAD
      CALL LEWIS(PHI,DD1,TC,RF,PD,TN1,PT1,YT,AKT)
      SBND2=FTIP*PD/(WD*YT)
      AGBN2=SBND2*AKT
      RTP1=RA2*SIN(AA2)/SIN(AA1)
      OMEGA=RPM*2.*PI
      VLG=(OMEGA/RAT)*RA2*SIN(P+AA2)
      VLP=OMEGA*RTP1*SIN(P-AA1)
      PVF=SQA*(VLG-VLP)/12.
      RETURN
      END
C
      SUBROUTINE LEWIS(PHI,B,TC,RF,PD,TN,PHIA,Y,AK)
C
C   LEWIS CALCULATES THE LEWIS FORM FACTOR FOR A PINION TOOTH
C
C      PHI = THE PITCH LINE PRESSURE ANGLE IN DEGREES
C        B = THE PINION DEDENDUM IN INCHES
C       TC = THE PITCH CIRCLE TOOTH THICKNESS IN INCHES
C       RF = THE CUTTER TIP RADIUS IN INCHES
C       PD = THE DIAMETRAL PITCH IN 1./INCHES
C       TN = THE PINION TOOTH NUMBER
C     PHIA = THE PRESSURE ANGLE AT THE POINT OF CONTACT
C            FOR WHICH Y IS CALCULATED IN DEGREES
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
C       Y = THE LEWIS FORM FACTOR
C      AK = THE TOOTH FILLET STRESS CONCENTRATION FACTOR
C
      FINV(X)=SIN(X)/COS(X)-X
      PI=3.14159265
      NW = 0
      ACC=.001
      P=PHI*PI/180.0
      PHA=PHIA*PI/180.0
      RP=TN/(2.*PD)
      YB=B-RF
      COP=COS(P)
      SIP=SIN(P)
      DELTA=PI/(2.*PD)-TC/2.0-YB*SIP/COP-RF/COP
      RC=RP*COP/(COS(PHA-TC/(2.0*RP)-FINV(P)+FINV(PHA)))
      BETA=PI/TN-DELTA/RP
      ER=1.
      DTH=.01
      THETA=.00
      DO 3 I=1,500
      THETA=THETA+DTH
      XB=RP*THETA
      RB=SQRT(YB**2+XB**2)
      COA=COS(BETA+THETA)
      SIA=SIN(BETA+THETA)
      XE=(RP-YB)*SIA-XB*COA-(RF/RB)*(YB*SIA+XB*COA)
      YE=(RP-YB)*COA+XB*SIA+(RF/RB)*(XB*SIA-YB*COA)
      SLOPE=-((1.+(YB/XB)*(SIA/COA))/(YB/XB-SIA/COA))
      ERR=SLOPE+(2.*(RC-YE))/XE
      IF(ABS(ERR).LT.ACC)GO TO 4
      IF(ERR/ER.LT.0.)DTH=-DTH/2.
      ER=ERR
    3 CONTINUE
      WRITE (NW,1)
    1 FORMAT('0    ITERATION FOR THETA UNSUCCESSFUL')
      Y=0.
      AK=1.0
      GO TO 5
    4 Y=2.*((XE**2)/(RC-YE))*PD/3.
      AP=0.4583662*P
      AH=0.340-AP
      AL=0.316-AP
      AM=0.290+AP
      RAP=RF+YB*YB/(RP+YB)
      BH=RC-YE
      TR=(2.0*XE/RAP)**AL
      TY=(2.0*XE/BH)**AM
      AK=AH+TR*TY
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
5 RETURN
  END
C
  SUBROUTINE GLIFE(FORCE,WD,RHP,RHG,TNP,TNG,PINRPM,BC,B,P,REL,
1      LM,LP,LG,CYLP,CYLG)
C
C         SUBROUTINE TO CALCULATE THE LIFE OF A GEAR MESH
C
C  INPUTS
C
C         FORCE  = NORMAL TOOTH LOAD IN POUNDS
C         WD     = EFFECTIVE FACE WIDTH IN INCHES
C         RHP    = PINION CONTACT RADIUS OF CURVATURE IN INCHES
C         RHG    = GEAR CONTACT RADIUS OF CURVATURE IN INCHES
C         TNP    = NUMBER OF TEETH ON PINION AS A REAL VARIABLE
C         TNG    = NUMBER OF TEETH ON GEAR AS A REAL VARIABLE
C         PINRPM = PINION SPEED IN REVOLUTIONS PER MINUTE
C         BC     = MATERIAL STRENGTH CONSTANT IN PSI
C         B      = WEIBULL SLOPE
C         P      = LOAD LIFE FACTOR
C         REL    = DESIRED RELIABILITY AS A DECIMAL
C
C  OUTPUTS
C
C         LM     = MESH LIFE IN HOURS
C         LP     = PINION LIFE IN HOURS
C         LG     = GEAR LIFE IN HOURS
C         CYLP   = PINION LIFE IN MILLION ROTATIONS
C         CYLG   = GEAR LIFE IN MILLION ROTATIONS
C
C
  REAL LM,LP,LG
  CT = BC*WD/(1.0/RHP + 1.0/RHG)
  EX = 1.0/(B*P)
  CP = CT*(1.0/TNP)**EX
  CG = CT*(1.0/TNG)**EX
  PL10 = (CP/FORCE)**P
  GL10 = (CG/FORCE)**P
  BR = 1.0/B
  RATLF = (ALOG(REL)/ALOG(0.9))**BR
  CYLP = PL10*RATLF
  CYLG = GL10*RATLF
  LP = CYLP*1000000.0/(60.0*PINRPM)
  LG = CYLG*1000000.0*TNG/(60.0*PINRPM*TNP)
  LM = 1.0/((1.0/TP)**B + (1.0/TG)**B)**BR
  RETURN
  END
C
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
      SUBROUTINE DYNAM (FT,V,DL)
C
C         DYLOAD CALCULATES THE DYNAMIC LOAD OF THE MESHING OF
C         THE GEARS USING THE AGMA VELOCITY FACTOR
C
C         DL = THE DYNAMIC LOAD DUE TO THE MESHING
C
C     INPUTS:
C         FT      = THE TRANSMITTED LOAD (POUNDS)
C         V       = THE PITCH LINE VELOCITY (FT/MIN)
C
C     OUTPUT:
C         DL      = THE TOTAL DYNAMIC LOAD (POUNDS)
C
      QV = 9.0
      A = 50.0 + 56.0*(1.0 - (12.0 - QV)**(2.0/3.0))/4.0
      VF = ( A + SQRT(V) )/A
      DL = FT*VF
      RETURN
      END
C
      SUBROUTINE TEMPER(PD,PHI,TN1,TN2,WD,DL,V,TB,RMS,TF,CSN)
C
C         TEMPER IS A SUBROUTINE THAT CALCULATES THE FLASH
C         TEMPERATURE, THE CRITICAL SCORING NUMBER AND THE MINIMUM
C         ELASTOHYDRODYNAMIC FILM THICKNESS FOR TWO GEARS IN MESH
C
C         PHI         = PRESSURE ANGLE IN DEGREES
C         TN1,TN2     = NUMBER OF TEETH ON PINION AND GEAR
C         PD          = DIAMETRAL PITCH
C         V           = PITCH LINE VELOCITY (FEET/MINUTE)
C         RMS         = TOOTH SURFACE FINISH (RMS)
C         RA2         = GEAR ADDENDUM RADIUS
C         RB2         = GEAR BASE RADIUS
C         WD          = FACE WIDTH
C         DL          = TOOTH DYNAMIC LOAD (POUNDS)
C         TB          = INLET OIL TEMPERATURE
C         C2          = RADIUS OF CURVATURE AT TIP OF GEAR TOOTH
C         TF          = FLASH TEMPERATURE
C         FT          = NORMAL TOOTH LOAD IN POUNDS
C
      PI = 3.14159265
      P   = PHI*PI/180.0
      COP = COS(P)
      SIP = SIN(P)
      R1 = TN1/(2.0*PD)
      R2 = TN2/(2.0*PD)
      C = R1 + R2
```

Table 18 Continued

Spur Gear Mesh Constraint and Objective Function Evaluation Routines

```
        GR = TN2/TN1
        RPM = 6.0*V/(PI*R1*COP)
        RA1 = R1 + 1.0/PD
        RB1 = R1*COP
        C1 = SQRT(RA1*RA1 - RB1*RB1)
        TAT = C1/RB1
        TAP = SIP/COP
        GY = TAT/TAP - 1.0
C
C            FD IS THE NORMAL FORCE ON THE GEAR
C
        FD = DL/3.0
C
C            CRITICAL SCORING TEMPERATURE OF THE GEAR
C
        XM = 1.75
        AX = ABS(SQRT(1.0 + GY) - SQRT(1.0 - GY/GR))
        BX = ((1.0 + GY)*(GR - GY))**0.25
        XG = 0.51*SQRT(GR + 1.0)*AX/BX
        SW = (FD/WD)**(0.75)
        TI = SW*(0.027*50.0/(50.0 - RMS))*XM*XG*SQRT(V)/(C**0.25)
        TF = TB + TI
C
C            CRITICAL SCORING NUMBER
C
        CSN = SW*SQRT(RPM)/(PD**.25)
        RETURN
        END
C
        SUBROUTINE GEARWT (RHO1,RHO2,R1,R2,WD,PW,GW)
C
C            GEARWT CALCULATES THE WEIGHT OF THE PINION AND GEAR
C            IN MESH
C
C            RHO1,RHO2   = MATERIAL DENSITY OF THE RESPECTIVE GEARS
C            R1,R2       = PITCH RADIUS OF THE PINION AND GEAR
C            WD          = FACE WIDTH OF GEARS
C            PW          = PINION WEIGHT
C            GW          = GEAR WEIGHT
C
        PI = 3.14159265
        PW = RHO1*PI*R1**2 *WD
        GW = RHO2*PI*R2**2 *WD
        RETURN
        END
C
```

pressure times velocity at the gear tip to model potential scoring. In order to calculate the bending stresses, subroutine MESH calls subroutine LEWIS which performs an interval halving iteration to determine the size of the largest inscribed parabola in the tooth. Subroutine GLIFE performs a Weibull analysis with equations (33) through (37) to estimate the life of the gear mesh for the reliability specified in the constant list. Subroutine GEARWT estimates the weights of the pinion and the gear and subroutine TEMPER calculates the flash temperature for the mesh using equation (32) to determine the potential for scoring damage.

Any permissible subroutine names may be used in programming for a specific problem as long as the names of the subroutines in SEEK are avoided. These names are listed in the file SEEK.DOC, which is Appendix A, and in Table 1 of the programming section and are: BACK, BOUNCE, BOUNDS, CHECK, GRADNT, MERIT, RESIZE, SCAN, SCOUT, SIZE, UNIT, VALUES and WALL. Once the problem subroutines have been written, they need to be compiled and linked to the compiled program SEEK.OBJ to produce an executable program with a name such as GEAR.EXE. With this program, various optimal gear designs can be found using different input files.

Numerical Example

For an example, let us consider the design of a gear set to transmit 10 horsepower from a shaft turning at 4,500 RPM to an output shaft turning at 3,000 RPM. The center distance of the gears should be minimized for a mesh life of at least 2,000 hours with a reliability of 90 percent. The face width ratio is to be less than or equal to 0.5, the material strengths are 40,000 psi in bending and 150,000 psi surface endurance, the PV factor limit

is 100 million psi-ft/min and the flash temperature limit is 275 degrees Fahrenheit.

Table 19 is a listing of the input file defining this example. Rules for writing the input data file are given in "SEEK.DOC," which is listed in Appendix A. The data file starts with the problem title. On the next line is the number thirteen, and the following thirty-nine lines contain the problem constants as listed in Table 17, their definitions and their units. The material surface constant of 9,800 psi corresponds to a surface compression endurance strength of 200,000 psi at 10^7 fatigue cycles and a reliability of 90 percent. The load-life factor of 8.93 is from the ANSI/AGMA 2001 B88 Standard, and the Weibull slope of 2.5 is from the NASA Lewis gear test data [21]. Following the constants is a line with the number three for the number of independent design parameters, which are on the next nine lines with their ranges, initial values, definitions and units. The number fourteen follows on the next line, with the fourteen constraint limits, their directions and their names and units on the following forty-two lines. After these comes a line with the optimization direction, MIN, and a line with the number two for the number of objective function terms. The first term is the center distance which has a weighting coefficient of one and the units of inches, and the second is the mesh life with a weighting coefficient of zero and units of thousand hours.

Running the program GEAR.EXE with this data file produced the output data file listed in Table 20. The first page lists the problem constants, design parameters and constraint limits as provided by the input data file but in a little more readable form. The second page lists the merit function terms and notes that an optimum was found in thirty-three steps. It then

93

Table 19

Spur Gear Mesh Input File for Minimum Size

```
COMPACT SPUR GEAR WITH A REQUIRED LIFE
13
0.25
POISSON'S RATIO

30000000.0
ELASTIC MODULUS
PSI
20.0
PRESSURE ANGLE
DEGREES
1.5
GEAR RATIO

10.0
TRANSMITTED POWER
HORSEPOWER
4500.0
PINION SPEED
RPM
0.283
MATERIAL WEIGHT DENSITY
LBS/IN**3
9800.0
MATERIAL SURFACE CONSTANT
PSI
2.5
WEIBULL SLOPE

8.93
LOAD-LIFE FACTOR

0.90
RELIABILITY

120.0
BASE TEMPERATURE
DEGREES F
32
TOOTH SURFACE FINISH
RMS
```

Table 19 Continued

Spur Gear Mesh Input File for Minimum Size

```
3
10.0 100.0 40.0
PINION TEETH

4.0 28.0 10.0
DIAMETRAL PITCH
1.0/INCHES
0.5 5.0 2.5
FACE WIDTH
INCHES
14
LOWER 0.001
INVOL. INTERFERENCE
RADIANS
UPPER 0.5
FACE WIDTH TO DIAMETER
RATIO
LOWER 0.0
CENTER DISTANCE
INCHES
LOWER 0.0
GEAR AND PINION WEIGHT
POUNDS
LOWER 0.0
PINION TORQUE
LB-IN
LOWER 0.0
TRANSMITTED FORCE
POUNDS
LOWER 0.0
PITCH LINE VELOCITY
FT/MIN
LOWER 0.0
TOTAL DYNAMIC LOAD
POUNDS
UPPER 40000.0
AGMA BENDING STRESS
PSI
UPPER 150000.0
FULL LOAD CONTACT STRESS
PSI
UPPER 150000.0
GEAR TIP HERTZ PRESSURE
PSI
UPPER 100.0
PV FACTOR
M PSI-FT/MIN
```

Table 19 Continued

Spur Gear Mesh Input File for Minimum Size

```
UPPER 275.0
FLASH TEMPERATURE
DEGREES F
LOWER 2.0
MESH LIFE
10**3 HOURS
MIN
2
1.0
CENTER DISTANCE
INCHES
0.0
MESH LIFE
10**3 HOURS
```

Table 20

Spur Gear Mesh Output File for Minimum Size

COMPACT SPUR GEAR WITH A REQUIRED LIFE

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1	POISSON'S RATIO	0.25000	
2	ELASTIC MODULUS	30000000.00000	PSI
3	PRESSURE ANGLE	20.00000	DEGREES
4	GEAR RATIO	1.50000	
5	TRANSMITTED POWER	10.00000	HORSEPOWER
6	PINION SPEED	4500.00000	RPM
7	MATERIAL WEIGHT DENSITY	0.28300	LBS/IN**3
8	MATERIAL SURFACE CONSTANT	9800.00000	PSI
9	WEIBULL SLOPE	2.50000	
10	LOAD-LIFE FACTOR	8.93000	
11	RELIABILITY	0.90000	
12	BASE TEMPERATURE	120.00000	DEGREES F
13	TOOTH SURFACE FINISH	32.00000	RMS

THERE ARE 3 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

	LOW	HIGH	INITIAL	
1 PINION TEETH	10.00000	100.00000	40.00000	
2 DIAMETRAL PITCH	4.00000	28.00000	10.00000	1.0/INCHES
3 FACE WIDTH	0.50000	5.00000	2.50000	INCHES

THE 14 CONSTRAINT LIMITS ARE:

				TYPE
1	INVOL. INTERFERENCE	0.00100	RADIANS	LOWER
2	FACE WIDTH TO DIAMETER	0.50000	RATIO	UPPER
3	CENTER DISTANCE	0.00000	INCHES	LOWER
4	GEAR AND PINION WEIGHT	0.00000	POUNDS	LOWER
5	PINION TORQUE	0.00000	LB-IN	LOWER
6	TRANSMITTED FORCE	0.00000	POUNDS	LOWER
7	PITCH LINE VELOCITY	0.00000	FT/MIN	LOWER
8	TOTAL DYNAMIC LOAD	0.00000	POUNDS	LOWER
9	AGMA BENDING STRESS	40000.00000	PSI	UPPER
10	FULL LOAD CONTACT STRESS	150000.00000	PSI	UPPER
11	GEAR TIP HERTZ PRESSURE	150000.00000	PSI	UPPER
12	PV FACTOR	100.00000	M PSI-FT/MIN	UPPER
13	FLASH TEMPERATURE	275.00000	DEGREES F	UPPER
14	MESH LIFE	2.00000	10**3 HOURS	LOWER

Table 20 Continued

Spur Gear Mesh Output File for Minimum Size

```
MINIMIZE THE OBJECTIVE FUNCTION, WHICH HAS  2 TERMS.

        OBJ = THE LINEAR SUM OF:

 1  CENTER DISTANCE                 IN  INCHES       TIMES    1.0000
 2  MESH LIFE                       IN  10**3 HOURS  TIMES    0.0000

        OPTIMIZATION SUCCESSFUL IN   33 STEPS

        THE FINAL DESIGN VECTOR IS:

                                            X(I)

 1  PINION TEETH                             20.39544
 2  DIAMETRAL PITCH                          13.91159  1.0/INCHES
 3  FACE WIDTH                                0.73295  INCHES

THE MINIMUM OBJECTIVE FUNCTION =    1.83259     ,  ITS COMPONENTS ARE:

 1  CENTER DISTANCE               = 1.8326      INCHES       TIMES    1.0000
 2  MESH LIFE                     = 2.0065      10**3 HOURS  TIMES    0.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION        = -0.326633E-04
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE =  0.976563E-04

        THE 14 CONSTRAINT VALUES ARE:
                                                            LIMIT       TYPE

 1 INVOL. INTERFERENCE        = .10861      RADIANS        .10000E-02  LOWER
 2 FACE WIDTH TO DIAMETER     = .49980      RATIO          .50000      UPPER
 3 CENTER DISTANCE            = 1.8326      INCHES         .00000      LOWER
 4 GEAR AND PINION WEIGHT     = 1.1376      POUNDS         .00000      LOWER
 5 PINION TORQUE              = 140.06      LB-IN          .00000      LOWER
 6 TRANSMITTED FORCE          = 191.07      POUNDS         .00000      LOWER
 7 PITCH LINE VELOCITY        = 1727.1      FT/MIN         .00000      LOWER
 8 TOTAL DYNAMIC LOAD         = 418.72      POUNDS         .00000      LOWER
 9 AGMA BENDING STRESS        = 25097.      PSI            40000.      UPPER
10 FULL LOAD CONTACT STRESS   = .14999E+06  PSI            .15000E+06  UPPER
11 GEAR TIP HERTZ PRESSURE    = .12518E+06  PSI            .15000E+06  UPPER
12 PV FACTOR                  = 92.633      M PSI-FT/MIN   100.00      UPPER
13 FLASH TEMPERATURE          = 216.49      DEGREES F      275.00      UPPER
14 MESH LIFE                  = 2.0006      10**3 HOURS    2.0000      LOWER
```

Spur Gear Mesh Output File for Minimum Size

DESIGN CHECK

X(I)

1	PINION TEETH	22.00000	
2	DIAMETRAL PITCH	14.00000	1.0/INCHES
3	FACE WIDTH	0.75000	INCHES

THE MINIMUM OBJECTIVE FUNCTION = 1.96429 , ITS COMPONENTS ARE:

1	CENTER DISTANCE	= 1.9643	INCHES	TIMES	1.0000
2	MESH LIFE	= 7.8155	10**3 HOURS	TIMES	0.0000

THE 14 CONSTRAINT VALUES ARE:

				LIMIT	TYPE
1	INVOL. INTERFERENCE	= .12489	RADIANS	.10000E-02	LOWER
2	FACE WIDTH TO DIAMETER	= .47727	RATIO	.50000	UPPER
3	CENTER DISTANCE	= 1.9643	INCHES	.00000	LOWER
4	GEAR AND PINION WEIGHT	= 1.3379	POUNDS	.00000	LOWER
5	PINION TORQUE	= 140.06	LB-IN	.00000	LOWER
6	TRANSMITTED FORCE	= 178.25	POUNDS	.00000	LOWER
7	PITCH LINE VELOCITY	= 1851.3	FT/MIN	.00000	LOWER
8	TOTAL DYNAMIC LOAD	= 398.15	POUNDS	.00000	LOWER
9	AGMA BENDING STRESS	= 22845.	PSI	40000.	UPPER
10	FULL LOAD CONTACT STRESS	= .13873E+06	PSI	.15000E+06	UPPER
11	GEAR TIP HERTZ PRESSURE	= .10980E+06	PSI	.15000E+06	UPPER
12	PV FACTOR	= 81.164	M PSI-FT/MIN	100.00	UPPER
13	FLASH TEMPERATURE	= 206.59	DEGREES F	275.00	UPPER
14	MESH LIFE	= 7.8155	10**3 HOURS	2.0000	LOWER

lists the found design parameter values, the objective function values and the constrained variable values with their limits. Although it is an "ideal" design with 20.395 teeth on the pinion, this theoretical optimum identifies the region of good designs. The smallest center distance for a life of 2,000 hours is about one and seven-eights inches and the limiting constraints were: the face width to diameter ratio, the full load contact stress and the mesh life. The watched variables in the constraint list tell us the weight of the design and its loads and velocities.

The third page is a design check for a design with 22 teeth on the pinion and a diametral pitch of 14 with a face width of 0.75 inches. This design has a slightly larger center distance of 1.964 inches, but all constraints are satisfied and the mesh life is about 7,800 hours. The gear and pinion weigh 1.34 pounds, the pinion torque is 140 pound-inches, the transmitted force is 178 pounds, and with a pitch line velocity of 1850 feet per minute, the dynamic load is estimated to be 400 pounds. Figure 15 is a drawing of this design showing the pinion and the gear with its 33 teeth in mesh.

With a few changes in the input data file, one can find a design with the greatest life for a given center distance. As can be seen in the second output file of Table 21, five changes were made in the input data file: 1) the problem title was changed, 2) the center distance limit was changed from a lower bound of zero to an upper bound of 2.5 inches, 3) the direction of optimization was changed from MIN to MAX, and 4 & 5) the two weighting factors in the objective function were switched to multiply the center distance objective term by zero and the mesh life objective term by one.

C = 1.964"

PINION
22 TEETH
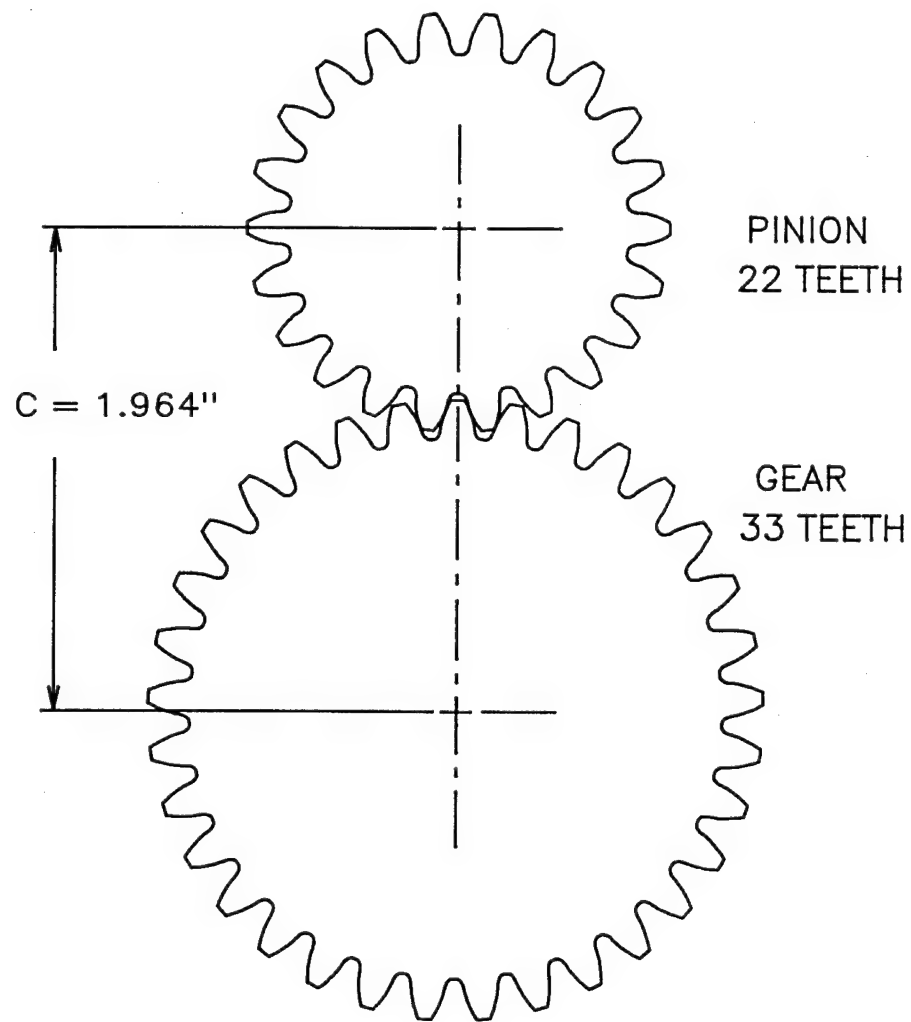
GEAR
33 TEETH

MINIMUM SIZE SPUR GEAR DESIGN

FIGURE 15

Table 21

Spur Gear Mesh Output File for Maximum Life

MAXIMUM LIFE SPUR GEAR WITH A FIXED SIZE

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1	POISSON'S RATIO	0.25000	
2	ELASTIC MODULUS	30000000.00000	PSI
3	PRESSURE ANGLE	20.00000	DEGREES
4	GEAR RATIO	1.50000	
5	TRANSMITTED POWER	10.00000	HORSEPOWER
6	PINION SPEED	4500.00000	RPM
7	MATERIAL WEIGHT DENSITY	0.28300	LBS/IN**3
8	MATERIAL SURFACE CONSTANT	9800.00000	PSI
9	WEIBULL SLOPE	2.50000	
10	LOAD-LIFE FACTOR	8.93000	
11	RELIABILITY	0.90000	
12	BASE TEMPERATURE	120.00000	DEGREES F
13	TOOTH SURFACE FINISH	32.00000	RMS

THERE ARE 3 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

	LOW	HIGH	INITIAL	
1 PINION TEETH	10.00000	100.00000	40.00000	
2 DIAMETRAL PITCH	4.00000	28.00000	10.00000	1.0/INCHES
3 FACE WIDTH	0.50000	5.00000	2.50000	INCHES

THE 14 CONSTRAINT LIMITS ARE:

				TYPE
1	INVOL. INTERFERENCE	0.00100	RADIANS	LOWER
2	FACE WIDTH TO DIAMETER	0.50000	RATIO	UPPER
3	CENTER DISTANCE	2.50000	INCHES	UPPER
4	GEAR AND PINION WEIGHT	0.00000	POUNDS	LOWER
5	PINION TORQUE	0.00000	LB-IN	LOWER
6	TRANSMITTED FORCE	0.00000	POUNDS	LOWER
7	PITCH LINE VELOCITY	0.00000	FT/MIN	LOWER
8	TOTAL DYNAMIC LOAD	0.00000	POUNDS	LOWER
9	AGMA BENDING STRESS	40000.00000	PSI	UPPER
10	FULL LOAD CONTACT STRESS	150000.00000	PSI	UPPER
11	GEAR TIP HERTZ PRESSURE	150000.00000	PSI	UPPER
12	PV FACTOR	100.00000	M PSI-FT/MIN	UPPER
13	FLASH TEMPERATURE	275.00000	DEGREES F	UPPER
14	MESH LIFE	0.00000	10**3 HOURS	LOWER

Table 21 Continued

Spur Gear Mesh Output File for Maximum Life

MAXIMIZE THE OBJECTIVE FUNCTION, WHICH HAS 2 TERMS.

OBJ = THE LINEAR SUM OF:

1	CENTER DISTANCE	IN	INCHES	TIMES	0.0000
2	MESH LIFE	IN	10**3 HOURS	TIMES	1.0000

OPTIMIZATION SUCCESSFUL IN   48 STEPS

THE FINAL DESIGN VECTOR IS:

X(I)

1	PINION TEETH	26.19858	
2	DIAMETRAL PITCH	13.09929	1.0/INCHES
3	FACE WIDTH	1.00000	INCHES

THE MAXIMUM OBJECTIVE FUNCTION =   4689.19   , ITS COMPONENTS ARE:

1	CENTER DISTANCE	= 2.5000	INCHES	TIMES	0.0000
2	MESH LIFE	= 4689.2	10**3 HOURS	TIMES	1.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION   = 0.976563E-03
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE = 0.953674E-07

THE 14 CONSTRAINT VALUES ARE:

				LIMIT	TYPE
1	INVOL. INTERFERENCE	= .15891	RADIANS	.10000E-02	LOWER
2	FACE WIDTH TO DIAMETER	= .50000	RATIO	.50000	UPPER
3	CENTER DISTANCE	= 2.5000	INCHES	2.5000	UPPER
4	GEAR AND PINION WEIGHT	= 2.8895	POUNDS	.00000	LOWER
5	PINION TORQUE	= 140.06	LB-IN	.00000	LOWER
6	TRANSMITTED FORCE	= 140.06	POUNDS	.00000	LOWER
7	PITCH LINE VELOCITY	= 2356.2	FT/MIN	.00000	LOWER
8	TOTAL DYNAMIC LOAD	= 334.97	POUNDS	.00000	LOWER
9	AGMA BENDING STRESS	= 12766.	PSI	40000.	UPPER
10	FULL LOAD CONTACT STRESS	= 96590.	PSI	.15000E+06	UPPER
11	GEAR TIP HERTZ PRESSURE	= 70072.	PSI	.15000E+06	UPPER
12	PV FACTOR	= 56.012	M PSI-FT/MIN	100.00	UPPER
13	FLASH TEMPERATURE	= 175.53	DEGREES F	275.00	UPPER
14	MESH LIFE	= 4689.2	10**3 HOURS	.00000	LOWER

103

Table 21 Continued

Spur Gear Mesh Output File for Maximum Life

DESIGN CHECK

X(I)

1	PINION TEETH	24.00000	
2	DIAMETRAL PITCH	12.00000	1.0/INCHES
3	FACE WIDTH	1.00000	INCHES

THE MAXIMUM OBJECTIVE FUNCTION = 4432.67 , ITS COMPONENTS ARE:

1	CENTER DISTANCE	= 2.5000	INCHES	TIMES	0.0000
2	MESH LIFE	= 4432.7	10**3 HOURS	TIMES	1.0000

THE 14 CONSTRAINT VALUES ARE:

#	Constraint	Value	Units	LIMIT	TYPE
1	INVOL. INTERFERENCE	= .14243	RADIANS	.10000E-02	LOWER
2	FACE WIDTH TO DIAMETER	= .50000	RATIO	.50000	UPPER
3	CENTER DISTANCE	= 2.5000	INCHES	2.5000	UPPER
4	GEAR AND PINION WEIGHT	= 2.8895	POUNDS	.00000	LOWER
5	PINION TORQUE	= 140.06	LB-IN	.00000	LOWER
6	TRANSMITTED FORCE	= 140.06	POUNDS	.00000	LOWER
7	PITCH LINE VELOCITY	= 2356.2	FT/MIN	.00000	LOWER
8	TOTAL DYNAMIC LOAD	= 334.97	POUNDS	.00000	LOWER
9	AGMA BENDING STRESS	= 12009.	PSI	40000.	UPPER
10	FULL LOAD CONTACT STRESS	= 97085.	PSI	.15000E+06	UPPER
11	GEAR TIP HERTZ PRESSURE	= 73216.	PSI	.15000E+06	UPPER
12	PV FACTOR	= 63.518	M PSI-FT/MIN	100.00	UPPER
13	FLASH TEMPERATURE	= 180.12	DEGREES F	275.00	UPPER
14	MESH LIFE	= 4432.7	10**3 HOURS	.00000	LOWER

For this new objective of maximizing the life of a larger gear set, the program found an optimum design with a life of 4.69 million hours in 48 steps. This design was bounded by the center distance limit of 2.5 inches and the length to diameter ratio limit of 0.5. This optimum also had an unrealistic number of teeth on the pinion of 26.2 and diametral pitch of 13.1. The last page of Table 21 shows the results of a nearby design check using 24 teeth on the pinion, a diametral pitch of 12 and a face width of one inch. This design is shown in Figure 16 with 24 teeth on the pinion and 36 teeth on the gear and the requested 2.5 inch center distance.

The realistic design has a slightly lower life of 4.43 million hours, a gear and pinion weight of 2.9 pounds, a pinion torque of 140 pound-inches, a transmitted force of 140 pounds, a pitch line velocity of 2356 feet per minute, and an estimated dynamic load of 335 pounds. Since it is larger than the minimum size design, it also has lower loads and stresses.

C = 2.5"

PINION
24 TEETH
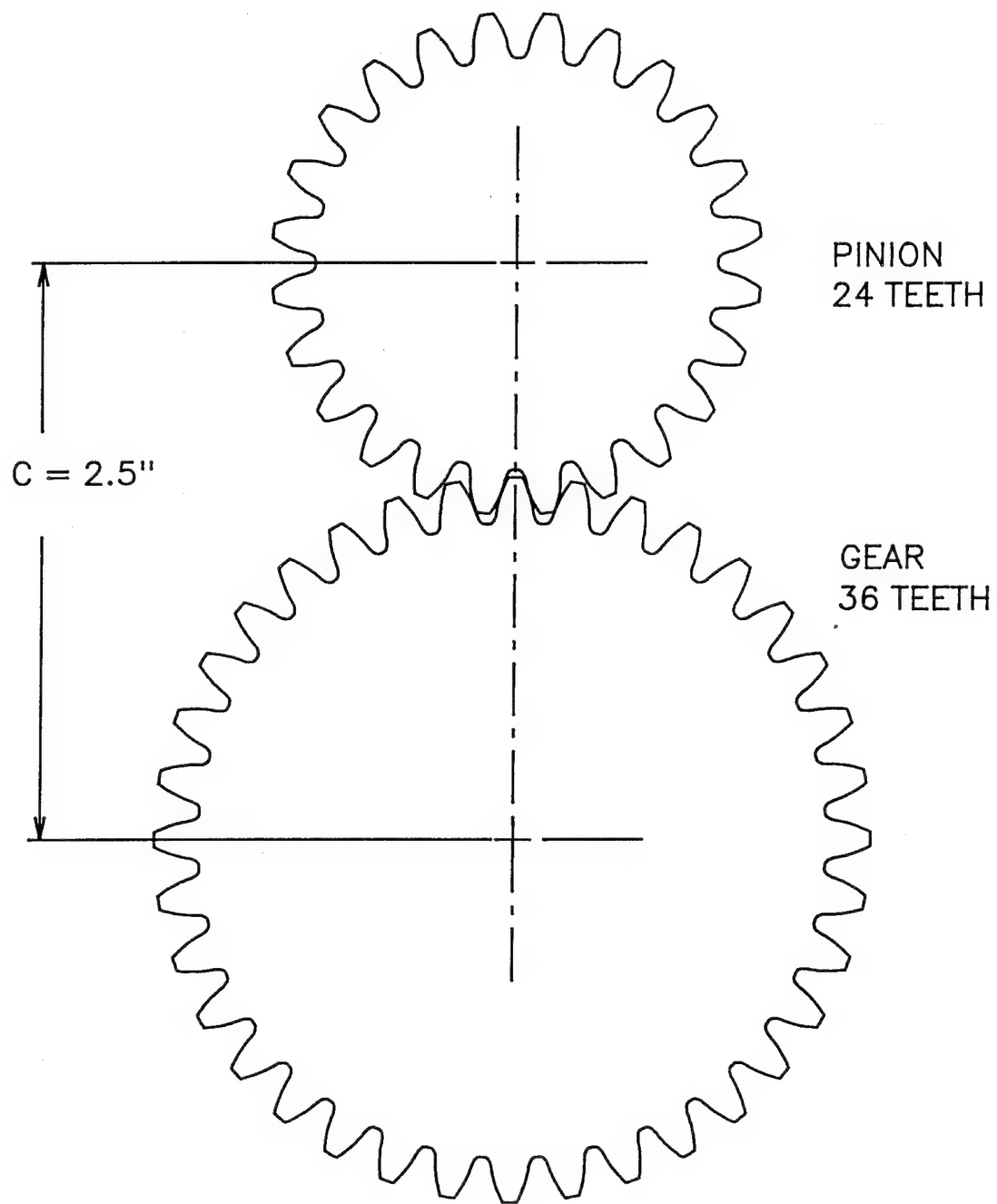
GEAR
36 TEETH

MAXIMUM LIFE SPUR GEAR DESIGN

FIGURE 16

WEIBULL DATA FITTING

The fourth example application is the use of the optimizer to fit experimental life data to a two-parameter Weibull distribution. With its abilities to iterate in n independent variables and provide an easy check of the performance of alternate solutions, the computer program SEEK.FOR offers a variety of uses. One valuable application is that of fitting a curve to experimental data.

Theory

A nonlinear least squares fitting procedure compares the measured data to the fitted curve at each point. The sum of the squares of the errors between the measured data and the values on the fitted curve is a positive definite scalar measure of the scatter of the data about the curve. Taking the square root of this squared error sum divided by the number of data points less one gives a dispersion with the same dimension as the measured quantity [22]. For the two-parameter Weibull relationship of equation (33), this dispersion would be:

$$\sigma_D = \left(\frac{(R_C - R_D)^2}{N_D - 1} \right)^{1/2} \tag{38}$$

where R_C is the reliability on the curve, R_D is the median rank reliability of the measured life and N_D is the number of data points.

Thus one can use SEEK.FOR to determine the ninety-percent reliability life, ℓ_{10}, and Weibull slope, b, from a set of life test data. Table 22 summarizes this problem in the design optimization format with: the lone constant being the number of data points; the two design parameters, ℓ_{10} and

107

Table 22

Spur Gear Mesh Optimization Parameters

Constants	Design Parameters	Inequality Constraints	Objective Function
N_D	ℓ_{10} b	$N_D > 0$	$(\sigma_D)_{min}$

Table 23

Median Ranked Life Data

Life 10^3 Hours	Reliability
190.0	0.95484
250.0	0.89060
310.0	0.82568
340.0	0.76061
410.0	0.69548
450.0	0.63033
510.0	0.56517
550.0	0.50000
600.0	0.43483
670.0	0.36967
710.0	0.30452
770.0	0.23939
790.0	0.17432
830.0	0.10940
880.0	0.04516

b; the inequality constraint of having a positive number of data points; and the objective function of minimizing the dispersion, σ_D.

For a group of identical units subjected to an identical life test, different units will fail at different times. By recording the times to failure and ranking them from the shortest to the longest, one can use a median rank table [23] to list the median reliability and test life in a table or data file similar to the numerical values in Table 23. This table lists the time to failure and the corresponding median reliability for a life test of fifteen units.

Programming

Inside SEEK.FOR is the additional capability to read a second data file in addition to the input data file. This action is enabled by setting the first constant in the constant array to the number, n, of data point pairs, (XD_i, YD_i), in the data file and including the words, "DATA POINTS," in its description. When this is combined with the presence of a data file of n lines with n point pairs; XD_i, YD_i; and a name which has the same prefix as the ".IN" file but the extension ".DAT;" SEEK will open and read the ".DAT" file after it has read the ".IN" file. The data pairs will be placed in a common block, COMMON/CURVE/XDP(200),YDP(200), for use in subroutine VALUES. If the limit of 200 data pairs is too low, changing this dimension in the program will allow larger data sets to be fit.

Table 24 lists subroutines BOUNDS and VALUES which will fit a two-parameter Weibull distribution to the data in the ".DAT" file. Subroutine BOUNDS checks that the number of data points is positive, since it has to do something to permit the overall program to execute. Subroutine VALUES uses

109

Table 24

Two-Parameter Weibull Fitting Analysis Subroutines

```
      SUBROUTINE BOUNDS(CONST,NCO,X,NX,VCSTR,NCS)
C
C         INPUT
C            CONST(1) = THE NUMBER OF DATA POINTS
C
C         OUTPUT
C            VCSTR(1) = THE NUMBER OF DATA POINTS
C
      DIMENSION X(NX),CONST(NCO),VCSTR(NCS)
      VCSTR(1) = CONST(1)
      RETURN
      END
C
      SUBROUTINE VALUES(CONST,NCO,X,NX,OBJECT,NOB)
C
C            MINIMIZE THE SUM OF THE FITTED ERRORS SQUARED
C
C         INPUTS
C            CONST(1) = THE NUMBER OF DATA POINTS
C
C            X(1) = RL - THE SOUGHT 90% RELIABILITY LIFE
C                                   OF THE DISTRIBUTION
C            X(2) = B - THE SOUGHT WEIBULL SLOPE
C
C         OUTPUT
C            OBJECT(1) = THE DEVIATION IN THE ERRORS
C
      DIMENSION X(NX),CONST(NCO),OBJECT(NOB)
      COMMON/CURVE/TDP(200),RDP(200)
      N = CONST(1)
      RL = X(1)
      B = X(2)
      ERS = 0.0
      DO 10 I = 1,N
      R = 1.0/EXP(ALOG(1.0/0.9)*(TDP(I)/RL)**B)
      E = R - RDP(I)
      ERS = ERS + E*E
   10 CONTINUE
      XN = N - 1
      OBJECT(1) = SQRT(ERS/XN)
      RETURN
      END
C
```

equations (33) and (38) to determine the dispersion of the measured data about the trial distribution.

Numerical Example

To illustrate this data fitting process, the fifteen data points of Table 23 and the analysis subroutines of Table 24 will be used. A listing of the input data file "W.IN" which will initiate the curve fitting is given in Table 25. In this file; the first line is the problem title; the second line is the number of constants, which is one; and the next three lines are the number of data points, 15; the description "NUMBER OF DATA POINTS" which includes the words "DATA POINTS" and a blank line for the units. Following this is a line with the number of independent parameters, two. The next six lines give the two distribution parameters, with their low, high and initial estimates. This is followed by the number of design constraints, one, and three lines which describe the lower bound on the number of data points. On the next line are the three letters "MIN" to signal that the objective function is to be minimized and four lines which give the number of object functions as one and the weighting coefficient, description and units for the reliability error dispersion. Once again, the unit line is blank.

Compiling the subroutines of Table 24, linking them with SEEK.OBJ, and running the resulting program with the two data files W.DAT which holds the numbers of Table 23 and W.IN which is listed in Table 25 yields the output file of Table 26. This file echoes the input file information and reports a successful curve fit in 43 steps. The fitted curve has an ℓ_{10} life of 238.8 thousand hours and a Weibull slope of 2.296 with a dispersion reliability of 0.0317.

111

Table 25

Two-Parameter Weibull Fitting Input Data File

```
WEIBULL RELIABILITY DISTRIBUTION FIT
1
15
NUMBER OF DATA POINTS

2
100.0 2000.0 500.0
90% RELIABILITY LIFE
10**3 HOURS
0.0 6.0 2.0
WEIBULL SLOPE

1
LOWER 0.0
NO. OF DATA POINTS

MIN
1
1.0
RELIABILITY ERROR DISPERSION
```

Table 26

Two-Parameter Weibull Fitting Output Data File

WEIBULL RELIABILITY DISTRIBUTION FIT

DESIGN WITH MODIFIED GRADIENT OPTIMIZATION
USING A MAXIMUM STEP LIMIT AND SCALED VARIABLES.

FIXED DESIGN REQUIREMENTS:

1   NUMBER OF DATA POINTS   15.00000

THERE ARE 2 INDEPENDENT DESIGN VARIABLES.

ESTIMATED VALUES:

		LOW	HIGH	INITIAL	
1	90% RELIABILITY LIFE	100.0000	2000.0000	500.0000	10**3 HOURS
2	WEIBULL SLOPE	0.0000	6.0000	2.0000	

THE 1 CONSTRAINT LIMITS ARE:

			TYPE
1	NO. OF DATA POINTS	0.00000	LOWER

MINIMIZE THE OBJECTIVE FUNCTION.

OBJ = RELIABILITY ERROR DISPERSION   IN   TIMES   1.0000

    15 POINTS READ FROM TABULAR DATA FILE

    OPTIMIZATION SUCCESSFUL IN   43 STEPS

    THE FINAL DESIGN VECTOR IS:

                                    X(I)

1	90% RELIABILITY LIFE	238.80429	10**3 HOURS
2	WEIBULL SLOPE	2.29637	

THE MINIMUM OBJECTIVE FUNCTION = 0.316652E-01 , ITS COMPONENTS ARE:

1   RELIABILITY ERROR DISPERSION =0.31665E-01   TIMES   1.0000

THE LAST CHANGE IN THE OBJECTIVE FUNCTION   = -0.372529E-08
THE LAST STEP CHANGE SIZE FOR THE DESIGN VARIABLE = 0.610352E-05

Table 26 Continued

Two-Parameter Weibull Fitting Output Data File

THE 1 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 NO. OF DATA POINTS	= 15.000		.00000	LOWER

DESIGN CHECK

                                            X(I)

		X(I)	
1	90% RELIABILITY LIFE	240.00000	10**3 HOURS
2	WEIBULL SLOPE	2.30000	

THE MINIMUM OBJECTIVE FUNCTION = 0.315955E-01 , ITS COMPONENTS ARE:

1 RELIABILITY ERROR DISPERSION =0.31596E-01   TIMES   1.0000

THE 1 CONSTRAINT VALUES ARE:

			LIMIT	TYPE
1 NO. OF DATA POINTS	= 15.000		.00000	LOWER

114

As shown in the design check, rounding the ℓ_{10} life to 240 thousand hours and the Weibull slope to 2.3 changes this dispersion to 0.0316. This rounded function fits the data slightly better than the "ideal" function as found by the optimizer due to the finite step size procedure of the gradient algorithm. Selecting a different starting position or sensitivity ranges for the two Weibull parameters could result in an "ideal" solution which is slightly better than the rounded solution, but it would not be significantly different. A good fit is achieved with two digit precision parameters, using the combination of the optimizer and the design check.

Figure 17 is a two-parameter Weibull graph of the measured data with its median rank reliability plotted as crosses. On this plot is drawn the rounded two-parameter Weibull distribution which has been fit to the data. It is always a good idea to draw the fitted curve with the data points to which it is fit. Since the dispersion is a single value representing the overall fit of the curve, some local anomalies may exist in a fitted curve. A graph shows these. If the data does not have the general shape of the fitted function, either a partial fit in the region of interest or a more sophisticated function should be used.

By changing the ranking from median to both low and high reliability ranks, the program can be run twice more to determine confidence bounds on the distribution. These two runs would give the user a low reliability fit to the test data and a high reliability fit to the same data. Plotting these two curves on the graph of Figure 17 would show the confidence range for the distribution.

To investigate the potential improvement in fitting the data with a higher order Weibull distribution, a three-parameter Weibull distribution

RELIABILITY — R (%)

LIFE — ℓ (10^3 HOURS)
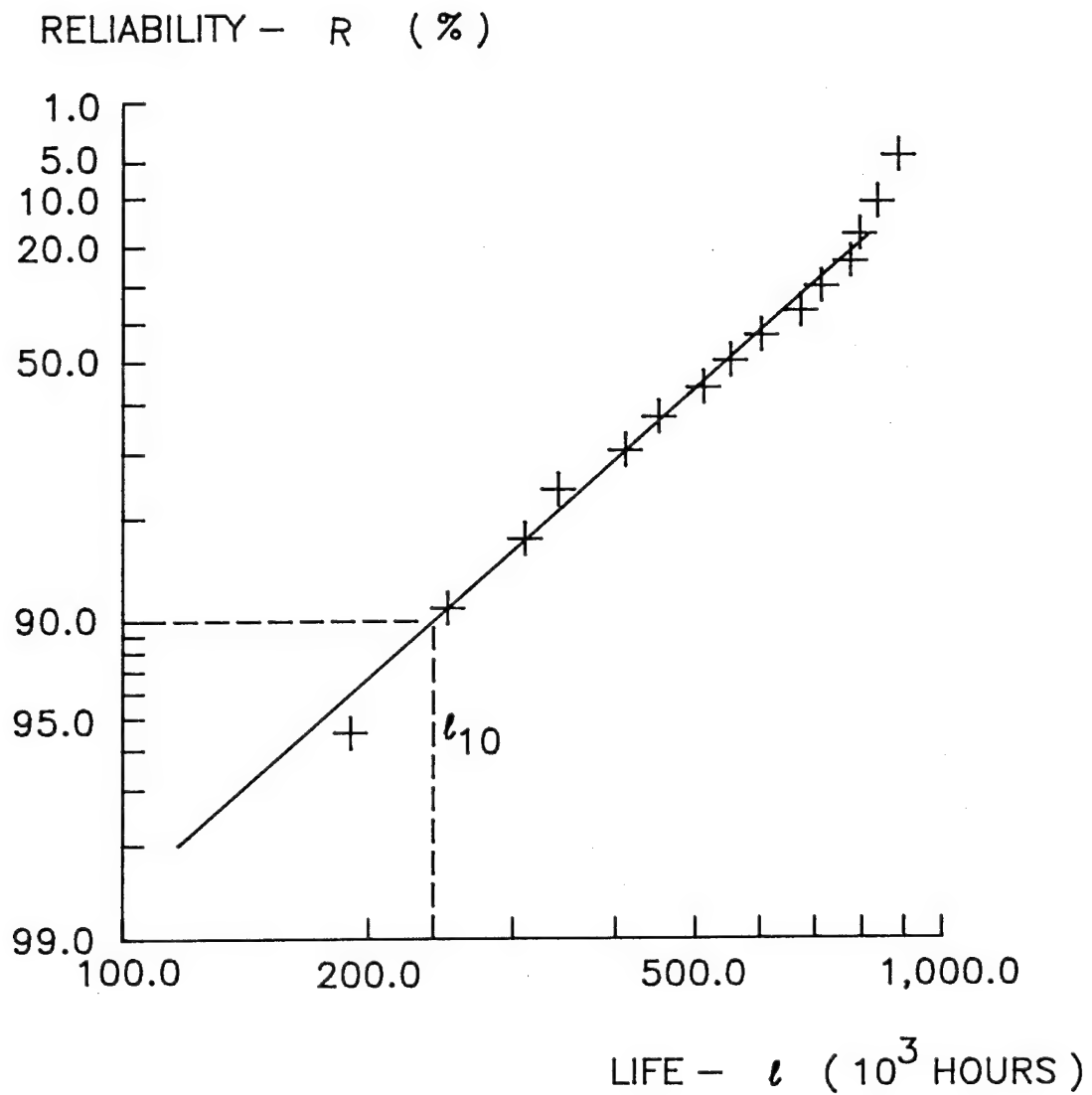
TWO—PARAMETER WEIBULL DISTRIBUTION
WITH ℓ_{10}= 240 ∗ 10^3 HOURS AND b = 2.3

FIGURE 17

could be fit to the same data by changing the reliability equation and adding a third independent parameter to the optimization. The equation for the three parameter Weibull distribution is:

$$\text{Ln}\left(\frac{1}{R}\right) = \text{Ln}\left(\frac{1}{0.9}\right) \cdot \left(\frac{\ell - \ell_0}{\ell_{10} - \ell_0}\right)^b \tag{39}$$

where the minimum life, ℓ_0, is the additional independent parameter. The third parameter, ℓ_0, replaces zero as the minimum life that all units would realize.

Modifying the program and the input data file would enable one to obtain a three-parameter Weibull distribution fit to the same data. Comparing the two dispersions would indicate whether the additional complexity of the three-parameter distribution is justified in the life model. A small minimum life would confirm the adequacy of the two-parameter Weibull distribution as well.

OPTIMIZATION METHOD

Parameter Scaling

As illustrated in the examples, the optimization procedure begins with several vectors. An optimization solution is the design variable vector, X, which minimizes or maximizes the objective function value, M, with all constraint values, V_k, bounded by their specified limits. A procedure starts with a guess for the design variables, X, and iterates using gradients to find the optimal values. Opportunity is then given to the user to try alternate solutions and compare their properties.

To maintain balance among the independent design parameters, the design space is scaled into a dimensionless design space [3]. The scaled design parameters, Y_i, vary from -1.0 to $+1.0$ as specified by upper and lower bounds on the independent design parameters, X_i, such that:

$$-1.0 < Y_i < +1.0 \tag{40}$$

as

$$X_{Li} < X_i < X_{Ui} \tag{41}$$

This linear transformation, which is shown in Figure 18, is:

$$Y_i = d_i X_i + b_i \tag{42}$$

where

$$d_i = \frac{2}{X_{Ui} - X_{Li}} \tag{43}$$

and

$$b_i = - \frac{X_{Ui} + X_{Li}}{X_{Ui} - X_{Li}} \tag{44}$$

118
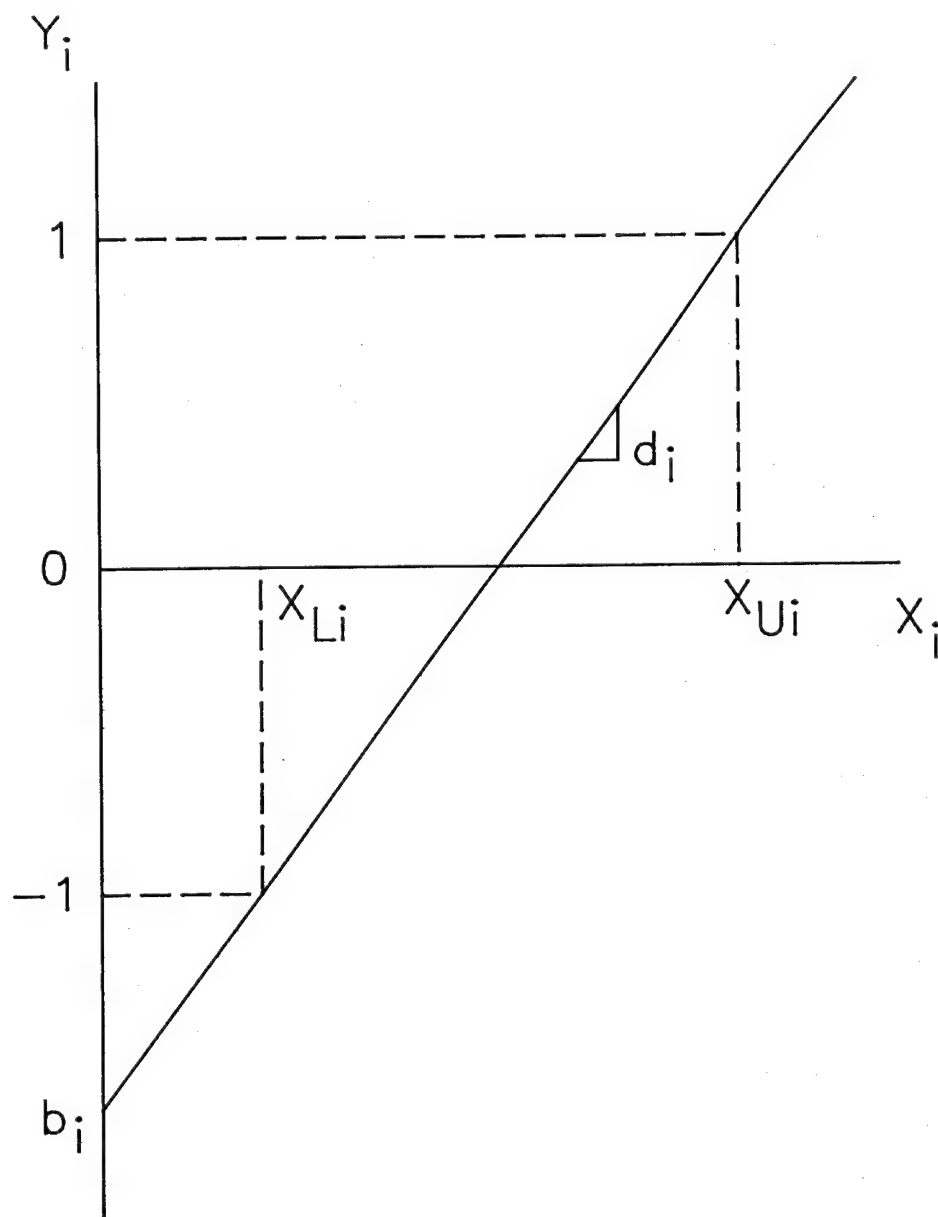
DESIGN PARAMETER SCALING
TRANSFORMATION

FIGURE 18

119

The actual design variable, X_i, can be retrieved from the scaled variable, Y_i, by the inverse transformation:

$$X_i = \frac{Y_i - b_i}{d_i} \tag{45}$$

Gradients

Central to this method is the gradient calculation which is performed with small perturbations in the design variables from the nominal position. The gradient in the merit function, ∇M, is calculated [1] as:

$$\nabla M = \begin{bmatrix} \dfrac{\partial M}{\partial Y_1} \\[2ex] \dfrac{\partial M}{\partial Y_2} \\[2ex] \dfrac{\partial M}{\partial Y_3} \\[1ex] \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{46}$$

where,

$$\frac{\partial M}{\partial Y_i} = \frac{M(Y_1, \cdot, Y_i + \Delta Y, \cdot, Y_n) - M(Y_1, \cdot, Y_i, \cdot, Y_n)}{\Delta Y} \tag{47}$$

In the program, the small change ΔY, which is made in each Y_i, is set at 0.001 which is 0.05 percent of the full range of a scaled design parameter.

The magnitude of the gradient vector is given by:

$$|\nabla M| = \left(\sum_{i=1}^{n} \left(\frac{\partial M}{\partial Y_i} \right)^2 \right)^{1/2} \qquad (48)$$

In the program, the step size is separated from the gradient magnitude but for minimization its direction is opposite to the gradient direction, so negative unit gradient vectors are determined:

$$\nabla m = - \frac{\nabla M}{|\nabla M|} \qquad (49)$$

For maximization, the sign in equation (49) reverses. This sign reversal also occurs for the constraint gradient when the limit is a lower bound.

Search Directions

In the simple gradient method which is used in the acceptable design region, equation (49) defines the direction of change in the scaled design vector, Y_j.

$$Y_{j+1} = Y_j + \Delta S \cdot \nabla m \qquad (50)$$

If no constraints are violated, this will be the next value for Y_j in the search.

For stability and directness, a nominally fixed step size, ΔS, is used in this optimization. Initially, the step size is 0.1, which is five percent of the range of a single design parameter. Then, whenever a local minimum is reached or the search is trapped in a constraint corner, the procedure halves the step. To complete the search procedure, the program declares a solution

when the percent change in the merit function is less than a pre-set limit of 0.0001:

$$\left| \frac{M_{j+1} - M_j}{M_j} \right| < 0.0001 \qquad (51)$$

Two other modes of searching are used in the procedure: 1) poor guess correction when constraints are violated initially, and 2) feasible direction searching near constraint boundaries.

These modes are enabled with a second gradient. Just as one can calculate the gradient in the objective function, one also can calculate the gradient in a violated constraint variable:

$$\nabla v_k = - \frac{\nabla V_k}{| \nabla V_k |} \qquad (52)$$

For upper bound constraints, moving through the design space in the direction of ∇v_k will reduce the constraint value V_k. For lower bound constraints, a sign reversal in equation (52) produces an increase in the constraint value, V_k, for motion in the gradient direction. The vector sum of the gradients in the violated constraints, ∇h, is the second gradient of the algorithm:

$$\nabla h = \frac{\sum_k \nabla v_k}{\left| \sum_k \nabla v_k \right|} \qquad (53)$$

The gradient in the violated constraints, ∇h, points towards the acceptable design space from the unacceptable design space. By itself, it

enables the algorithm to turn an unacceptable initial guess into an acceptable design trial by a succession of steps:

$$Y_{j+1} = Y_j + \Delta S \cdot \nabla h \tag{54}$$

Once inside the acceptable design region, the algorithm proceeds along the steepest descent direction with equation (50) until the calculated step places the next trial outside the acceptable design space. To avoid this condition, the algorithm selects a feasible direction for the next step. Figure 19 shows a sloped constraint intersecting vertical contour lines of the objective function. This figure is an enlargement of a small region in Figure 2, which is a plot of the length versus diameter for a bushing. In this example, the objective function is directly proportional to the bushing diameter and independent of its length. So a steepest descent direction is always horizontal for the problem. The sloped constraint curve is the length to diameter upper bound. Figure 19 shows vertical contour lines in the frictional torque objective function and unit gradient vectors in this objective function, ∇m, and the impending constraint, ∇h. The two gradient vectors are added at the last viable design step. The feasible direction selected, ∇f, is the unit vector sum of these two gradients:

$$\nabla f = \frac{\nabla m + \nabla h}{| \nabla m + \nabla h |} \tag{55}$$

And the next step becomes:

$$Y_{j+1} = Y_j + \Delta S \cdot \nabla f \tag{56}$$

123
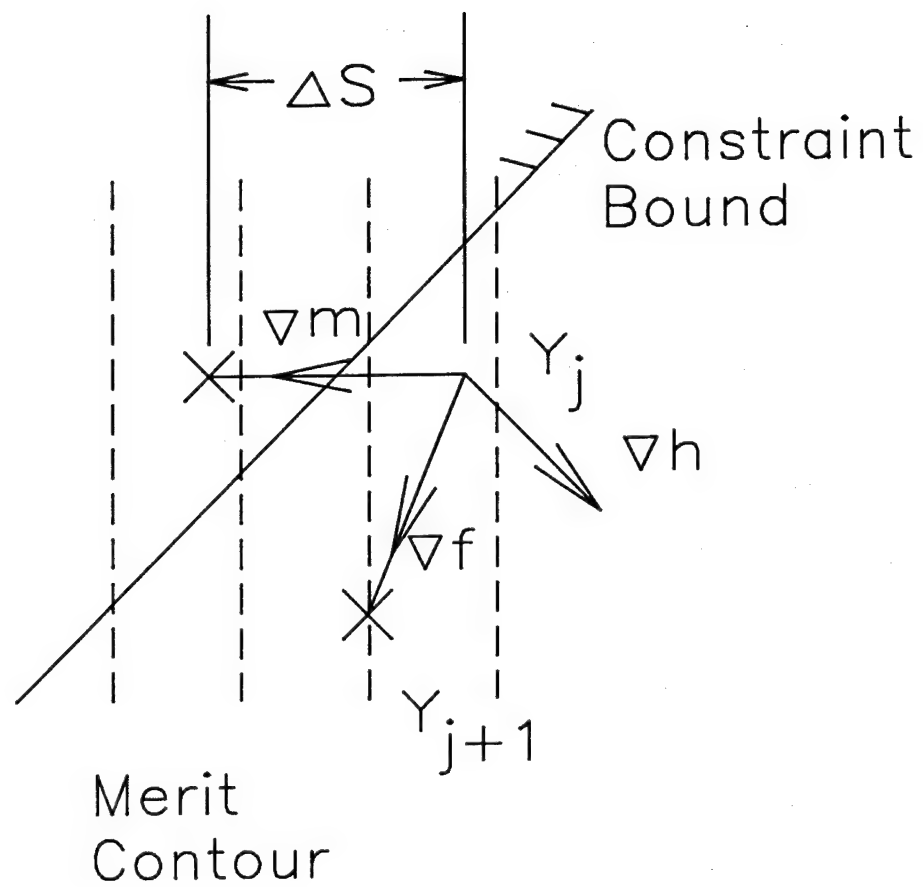
FEASIBLE DIRECTION GRADIENT

FIGURE 19

Program Structure

In writing the optimization code, subroutines were used to modularize the programming and perform the vector calculations in a structured way. Table 1, which is cited in the PROGRAMMING section of this report, lists the subroutines which compose this program with short descriptions of their functions. Each subroutine's name is related to its operation to describe its use.

The scaling and unscaling of the design parameter vector are performed by two routines: SIZE and RESIZE, so that this linear transformation can be performed anywhere in the program with relative ease. Subroutines GRADNT and UNIT evaluate and normalize any gradient vector. Subroutine BOUNCE finds the gradient sum of the violated constraints at any design position. Subroutine CHECK compares a constraint value to its limit and subroutine WALL evaluates a specific constraint in the same way that subroutine MERIT evaluates the objective function.

At a higher level in the program structure, subroutine BOUNCE directs the search for the acceptable design space when the initial design guess violates at least one constraint. Subroutine SCAN performs the search for better designs within the acceptable design space and subroutine SCOUT checks the next potential design for constraint violations. Finally, subroutines BOUNDS and VALUES are the user supplied, problem specific routines which evaluate the design constraints and the objective function values for the design.

Figure 20 is a logic flow chart for the optimization program which includes: 1) the reading of the input data files, 2) the echoing of the input data file information at the start of the output data file, 3) the initial

125

```
        ┌─────────────────────────────────────┐
        │ READ DATA & SET CONSTANTS           │
        │ WRITE INPUT DATA TO OUTPUT FILE     │
        └─────────────────────────────────────┘
                        │
        ┌─────────────────────────────────────┐
        │ FOR FIRST CONSTANT OF "DATA POINTS" │
        │ READ TABLE OF XDP, YDP VALUES       │
        └─────────────────────────────────────┘
                        │
        ┌─────────────────────────────────────────┐
        │ CHECK CONSTRAINT & MERIT FUNCTION VALUES │
        └─────────────────────────────────────────┘
                        │
        ┌──────────────────────────────┐   no   ┌──────────────────┐
        │ ARE ALL CONSTRAINTS SATISFIED?├───────→│ MOVE ALONG ∇h    │
        └──────────────────────────────┘        │ TO A NEW POINT   │
                        │ yes                    └──────────────────┘
        ┌──────────────────────────────────────────┐
        │ CALCULATE ∇m & LOCATION OF NEXT POINT     │
        │ ALONG ∇m & CONSTRAINT VALUE THERE         │
        └──────────────────────────────────────────┘
                        │
        ┌──────────────────────────────┐   no   ┌──────────────┐
        │ ARE ALL CONSTRAINTS SATISFIED?├───────→│ CALCULATE ∇f │
        └──────────────────────────────┘        └──────────────┘
                        │ yes                            │
                        │              ┌──────────────────────────────┐  no  ┌──────────┐
                        │              │ ARE ALL CONSTRAINTS           ├─────→│ ΔS = ΔS/2│
                        │              │ SATISFIED AT NEXT ∇f STEP ?   │      └──────────┘
                        │              └──────────────────────────────┘
                        │                       │ yes
        ┌────────────────────────────┐   ┌──────────────────────────┐
        │ TAKE STEP IN ∇m DIRECTION  │   │ TAKE STEP IN ∇f DIRECTION │
        └────────────────────────────┘   └──────────────────────────┘
                        │
        ┌──────────────────────────────┐   no   ┌──────────┐
        │ DID MERIT FUNCTION IMPROVE ?  ├───────→│ ΔS = ΔS/2│
        └──────────────────────────────┘        └──────────┘
                        │ yes
        ┌──────────────────────────────────────┐   no
        │ DID MERIT FUNCTION BECOME ZERO ?      ├──────→
        │ IS ITS PERCENT CHANGE SMALL ?         │
        │ OR HAVE 500 TRIALS BEEN MADE ?        │
        └──────────────────────────────────────┘
                        │ yes
        ┌──────────────────────────────────────┐
        │ WRITE FOUND DESIGN AND ITS PROPERTIES │
        └──────────────────────────────────────┘
                        │
        ┌──────────────────────────────┐   no   ┌──────┐
        │ OFFER CHANCE TO CHANGE DESIGN ├───────→│ STOP │
        └──────────────────────────────┘        └──────┘
                        │ yes
        ┌──────────────────────────────────┐
        │ ACCEPT NEW DESIGN PARAMETER VALUES│
        └──────────────────────────────────┘
                        │
        ┌──────────────────┐
        │ ANALYZE DESIGN   │
        └──────────────────┘
```

SEEK PROGRAM FLOW CHART

FIGURE 20

search for a good starting design, 4) the main optimization loop, and 5) the design verification loop.

The search for a good starting design which satisfies all constraints is performed in a DO loop with a limit of twenty corrections. If a valid design is not found in twenty trials, both the first and twentieth trials are written to the output file with an analysis of all constraints, the file is closed and the program is stopped.

Once a good starting design is found, the search for the optimal design is conducted in the main DO loop with a limit of five-hundred iterations. In this DO loop, subroutine SCOUT checks the constraint values at the next design point using the objective function gradient increment as shown in equation (50) to locate the next design point. If all the constraints are satisfied, the full design step is taken in the direction of the objective function gradient using equation (50). If at least one constraint is not satisfied, subroutine SCAN increments the design along the feasible direction gradient using equation (56). If all the constraints are still not satisfied, the step size is reduced and the half step is taken in the feasible direction.

After a step is taken, the objective function value is then checked. If it increased, the step size is divided by two. Otherwise, no change is made. Finally, the percent change in the objective function is checked using MERIT. If this change is below the desired limit, a solution is declared and the procedure leaves the DO loop. If the change is greater than the desired limit, the DO loop indexes and the iteration process is repeated.

If the merit function criteria are not satisfied after five-hundred iterations, the program writes the five-hundredth design to the screen and the output file with an error message and provides the user with the opportunity

127

to change the design and have it re-analyzed. This documentation and opportunity for modification are also given to the user at the end of a successful design optimization search. The final design modification loop is controlled by the user with no termination count. Additional designs may be evaluated until the user chooses to end the program.

DISCUSSION OF RESULTS

In this report, a computer program, 'SEEK.FOR,' is described and its use is illustrated with four different examples. SEEK has been written to make optimization available to the general technical community with only the need for modeling the applied problem. The program is a general purpose optimizing tool which can find the best combination of design parameter values which satisfy a series of constraints placed on calculable properties of the design parameters. The set of design parameter values is 'best' in that it maximizes or minimizes a linear sum of objective function terms.

Although the computer program requires the user to write two analysis subroutines and an input data file for his or her application, it is written to be interactive and to communicate clearly to the user. By requiring text labels for all variables, the program is able to label its restatement of the problem and all results in the words of the user. In the input file, the user is able to change the starting design values and the relative sensitivity of the individual design parameters by changing the low and high estimates of these values.

Based on a bounded step gradient method, the program searches with gradients in three functions: 1) the objective function, 2) the violated or nearly violated constraints and 3) a vector sum of the first two. The first gradient provides a rapid solution path when no obstacles are present. The second gradient permits the user to start with an initial trial design which may not satisfy all constraints. And the third gradient allows the algorithm to continue improving the objective function even when the direct path is blocked with one or more constraints.

To enable the gradient calculations, the provided models for the objective function terms and the constrained variables need to be continuous. Discrete values can be entered later in a design check mode which the program enters once a numerical optimum has been found in the continuous design space. This design check mode is part of the interactive operation of the program. By allowing the user to try alternate designs after the numerical optimum has been found, the program assists the user in finding practical optimums which satisfy discrete requirements which can not be treated in the continuous models. An output file is written during the session to document the design problem and all solutions which are displayed on the screen. In this file, each solution has its design parameter values, objective function values, and constrained variable values and limits listed with the users descriptions as labels.

This report illustrates the model preparation and programming required to use 'SEEK.FOR' for four examples: 1) a bushing design, 2) a spring design, 3) a gear design, and 4) a curve fit. The bushing design problem illustrates the determination of two parameters: the bushing length and its diameter subject to three inequality constraints, a single objective function and discrete size requirements. The example also shows the program's ability to overcome a poor initial trial design.

The spring design problem illustrates the conversion of equality constraints to simplify the problem by taking a four-parameter problem and treating it as a problem with two independent parameters. The number of active coils in the spring and the free length of the spring are shown to be dependent parameters and the wire diameter and mean coil diameter are left as independent parameters. The example finds three separate designs which all

130

have the same stiffness and design factor under the same loading but which satisfy three separate objective functions. By changing the objective function for the same design constraints, the program finds the lightest spring, the smallest volume spring and the shortest spring in separate runs. A visual comparison of the three springs shows their obvious differences.

The gear design problem illustrates a three parameter design problem which requires considerably more analysis than the first two. The three parameters are the number of teeth on the pinion, the diametral pitch and the face width of the gears. Designs are obtained for a minimum size at a given life and for a maximum life at a given size. The example illustrates the use of additional subroutines by the two analysis routines: BOUNDS and VALUES.

The curve fit problem illustrates the use of the optimizing program as a tool for minimizing the dispersion of the data from a modeled function. Although the program can minimize the overall error between a fitted curve and the experimental data, it is recommended that a plot of the fitted curve and the original data be made to verify that the function truly models the data closely in the region of highest interest. One feature of this use of the program is the ability in the design check mode to quickly see the influence of curve coefficients on the goodness of fit.

No optimization program can solve all problems. The use of this program is limited to solving problems which can be modeled continuously with finite constraint and objective function values over the search area. The program contains arbitrary size limits of forty for the constants and constraints and fifteen for the design parameters and objective function terms. As the number of independent design parameters increases from the low numbers of these examples, gradient searching becomes more difficult. More steps are required

131

to find a solution and more calculations are required for each step. Design parameters, which do not change, can dilute the effectiveness of the method with additional calculations. Often, a large optimization problem can be reduced to a simpler one which permits a more rapid and direct solution by the removal of design parameters which do not change from the design parameter vector.

For any problem which can be modeled continuously, the program 'SEEK.FOR' offers an easily used, interactive tool for the determination of a practical optimum solution. The program is small with an object code size of 46 k bytes and runs quickly on a 486 personal computer running DOS.

SUMMARY OF RESULTS

In this report, the use of a computer program, 'SEEK.FOR' is demonstrated with four optimization examples. The program is written to work with two user-provided subroutines and an input data file. Its purpose is to perform a gradient search optimization of a user's problem. The method of optimization uses a modified feasible directions gradient in addition to simpler gradients in the objective function and the violated constraints. The program is written in ANSI standard Fortran 77, is about 1,200 lines long and has an object size of about 46 K bytes for the optimizing code for use on a personal computer running DOS.

To illustrate the use of the program, this report documents four optimization examples: a bushing design, a helical coil spring design, a gear mesh design and a two-parameter Weibull life-reliability curve fit. In each example, the theory of the problem is described, the organization of the problem into an optimization framework is given, the programming of the modeling subroutines is explained and a numerical example is shown with the input and output data files explained.

In the bushing design problem, two independent design parameters are found which minimize a single objective function subject to three inequality constraints. In the spring design problem, four design parameters are reduced to two independent design parameters and three separate objective functions are minimized with three separate runs of the same program. The designs satisfy seven inequality constraints. For the gear design problem, two sets of three independent design parameters are found in two separate runs to satisfy two opposing objective functions. In this model, fourteen inequality constraints are satisfied. For the curve fitting example, a table of x and y

133

data points is read into the program and two parameters are found which fit a curve to the data with minimized dispersion. One innocuous inequality constraint is used to allow the program to run, since the program requires the subroutine to be in place. As written, the program accepts problems with up to forty input data constants and forty inequality constraints. The limit on the number of independent design parameters is fifteen as is the limit on the number of objective function terms.

This report describes the use of the optimizing program, gives four examples of its use and discusses the program and method themselves. The program 'SEEK.FOR' is an adaptable, analytical tool for finding optimal solutions to technical problems.

SEEK.DOC FILE

```
C
C              PROGRAM SEEK.FOR
C
C          MODIFIED GRADIENT OPTIMIZATION PROGRAM
C
C          WITH INEQUALITY CONSTRAINT AVOIDANCE
C
C          LINK THIS PROGRAM WITH TWO PROBLEM SPECIFIC ANALYSIS
C          SUBROUTINES AND RUN WITH A PROBLEM SPECIFIC DATA FILE.
C
C              "NAME".IN  DATA INPUT FILE REQUIRED
C
C                      FORMAT:
C
C          LINE 1                                : NTITLE (50 CHAR. MAX)
C                      FOR THE CONSTANTS
C          LINE 2                                : NCO
C                      IN SETS OF THREE LINES EACH
C          LINE 3 - (1A)                         : CONST(I)
C          LINE 4 - (2A)                         : NCON (30 CHAR. MAX)
C          LINE 5 - (3A)                         : NUCON (12 CHAR. MAX)
C                      THEN FOR THE INDEPENDENT VARIABLES
C          LINE 3 + 3*NCO                        : NX
C                      IN SETS OF THREE LINES EACH
C          LINE (1B)                             : XLOW(I) , XHIGH(I) , XZ(I)
C          LINE (2B)                             : NVAR (30 CHAR. MAX)
C          LINE (3B)                             : NUVAR (12 CHAR. MAX)
C                      THEN FOR THE CONSTRAINTS
C          LINE 4 + 3*NCO + 3*NX                 : NCS
C                      IN SETS OF THREE LINES EACH
C          LINE (1C)                             : LIMIT(I) , CSTR(I)
C          LINE (2C)                             : NCSTR (30 CHAR. MAX)
C          LINE (3C)                             : NUCSTR (12 CHAR. MAX)
C                      THEN FOR THE OBJECTIVE FUNCTION
C          LINE 5 + 3*NCO + 3*NX + 3*NCS         : DIR
C          LINE 6 + 3*NCO + 3*NX + 3*NCS         : NOB
C                      IN SETS OF THREE LINES EACH
C          LINE (1D)                             : WGHTF(I)
C          LINE (2D)                             : NOBJ (30 CHAR. MAX)
C          LINE (3D)                             : NUOBJ (12 CHAR. MAX)
C
C
C          TO HAVE ACCESS TO A TABLE OF X AND Y DATA VALUES,
C      LET THE FIRST CONSTANT BE THE NUMBER OF DATA POINTS
C      AND INCLUDE THE WORDS "DATA POINTS" IN ITS DESCRIPTION.
C
C          A DATA FILE WITH X,Y DATA PAIRS AND THE NAME "NAME".DAT
C      SHOULD ALSO EXIST IN THE SAME DIRECTORY.
```

135

```
C
C               UP TO 200 DATA PAIRS WILL BE PLACED IN THE COMMON BLOCK
C         COMMON /CURVE/XDP(200),YDP(200).
C
C                   TWO ANALYSIS SUBROUTINES ARE REQUIRED
C
C                   BOUNDS(CONST,NCO,X,NX,VCSTR,NCS)   ; AND
C                   VALUES(CONST,NCO,X,NX,OBJECT,NOB)
C
C                   BOUNDS EVALUATES THE INEQUALITY CONSTRAINTS
C
C                   VALUES DETERMINES THE OBJECTIVE FUNCTION'S
C                          COMPONENT PROPERTIES
C
C                       DEFINITIONS:
C
C         CONST  - FIXED DESIGN PROBLEM CONSTANT
C         CSTR   - CONSTRAINT LIMIT VALUE INCLUDING DECIMAL POINT
C         DIR    - OPTIMIZATION DIRECTION ( MIN , MAX )
C         LIMIT  - CONSTRAINT LIMIT BOUND TYPE ( UPPER , LOWER )
C         NCO    - NUMBER OF PROBLEM CONSTANTS
C         NCS    - NUMBER OF INEQUALITY CONSTRAINTS
C         NOB    - NUMBER OF PROPERTIES IN MERIT FUNCTION
C         NX     - NUMBER OF INDEPENDENT VARIABLES
C         NCON   - CONSTANT NAME
C         NUCON  - CONSTANT DIMENSION UNITS
C         NVAR   - VARIABLE NAME
C         NUVAR  - VARIABLE DIMENSION UNITS
C         NCSTR  - CONSTRAINT NAME
C         NUCSTR - CONSTRAINT DIMENSION UNITS
C         NOBJ   - MERIT FUNCTION COMPONENT NAME
C         NUOBJ  - MERIT FUNCTION COMPONENT DIMENSION UNITS
C         NTITLE - DESIGN PROBLEM TITLE
C         OBJECT - MERIT FUNCTION COMPONENT VALUES
C         VCSTR  - CONSTRAINT FUNCTION VALUES
C         WGHTF  - WEIGHTING COEFFICIENT FOR COMPONENT IN
C                      LINEAR MERIT FUNCTION SUM
C         X      - INDEPENDENT DESIGN VARIABLE
C         XHIGH  - HIGH VARIABLE VALUE
C         XLOW   - LOW VARIABLE VALUE
C         XZ     - INITIAL VARIABLE VALUE
C
C         AT PRESENT, THE ARRAY SIZE LIMITS ARE:
C
C                   MAX NCO = 40,
C                   MAX NCS = 40,
C                   MAX NOB = 15, AND
C                   MAX NX  = 15.
C
C
```

```
C          AVOID THE USE OF SUBROUTINE AND COMMON BLOCK NAMES
C     WHICH ARE ALREADY USED BY SEEK.
C
C          THE SUBROUTINE NAMES USED BY SEEK ARE:
C
C          BACK. . . . . . . . . . . . FSUBRT
C          BOUNCE. . . . . . . . . . . FSUBRT
C          BOUNDS. . . . . . . . . . . extern
C          CHECK . . . . . . . . . . . FSUBRT
C          GRADNT. . . . . . . . . . . FSUBRT
C          MERIT . . . . . . . . . . . FSUBRT
C          RESIZE. . . . . . . . . . . FSUBRT
C          SCAN. . . . . . . . . . . . FSUBRT
C          SCOUT . . . . . . . . . . . FSUBRT
C          SIZE. . . . . . . . . . . . FSUBRT
C          UNIT. . . . . . . . . . . . FSUBRT
C          VALUES. . . . . . . . . . . extern
C          WALL. . . . . . . . . . . . FSUBRT
C
C          THE COMMON BLOCK NAMES USED BY SEEK ARE:
C
C          CURVE . . . . . . . . . . . common
C          PAR . . . . . . . . . . . . common
C          VAR . . . . . . . . . . . . common
C          UNITS . . . . . . . . . . . common
C
C          TO PRINT OUT INTERMEDIATE RESULTS FROM BOUNDS, VALUES
C     OR ANY SUBROUTINE CALLED BY BOUNDS OR VALUES, INCLUDE THE
C     COMMON BLOCK "UNITS" IN THAT SUBROUTINE, AS FOLLOWS:
C
C     COMMON /UNITS/NW,NR,NF,ND
C
C     WHERE:
C          NW = WRITE NUMBER FOR WRITING TO THE SCREEN
C                    - WRITE(NW,...)
C          NR = READ NUMBER FOR READING FROM THE KEYBOARD
C                    - READ(NR,...)
C          NF = WRITE NUMBER FOR WRITING TO THE OUTPUT FILE
C                    - WRITE(NF,...) ,AND
C          ND = READ NUMBER FOR READING FROM THE INPUT DATA FILE
C                    - READ(ND,...).
C
C
```

APPENDIX B

SYMBOLS

Variables

A   – AGMA velocity factor constant

b_i – independent design variable scaling constant

B   – material surface strength constant (psi)

C   – center distance (in)

C_2 – minimum gear tooth curvature (in)

C   – dynamic capacity (lbs)

C_g – gear dynamic capacity (lbs)

C_t – tooth dynamic capacity (lbs)

d_i – independent design variable scaling slope

d_w – wire diameter (mm)

D   – shaft or mean coil diameter (mm)

E   – elastic modulus (psi)

f   – face width (in)

F   – applied force (N)

∇f – normalized feasible direction gradient

G   – shear modulus (MPa)

h_f – unloaded spring height (mm)

∇h – normalized violated constraint gradient

J   – polar moment of inertia (mm^4) or AGMA bending stress factor

k   – spring rate (N/mm)

K_w – Wahl stress concentration factor

ℓ – life (h)

ℓ_o – minimum life (h)

138

ℓ_m - mesh life (10^3 h)

L   - bearing length (mm)

M   - objective function vector

∇m - normalized objective function gradient

∇M - unscaled objective function gradient

N_a - number of active spring coils

N_{des} - design factor

N_D - number of data points

N_e - number of inactive end coils

N_f - design factor in fatigue

N_g - number of gear teeth

N_s - static design factor

OD   - outside coil diameter (mm)

P   - contact pressure (MPa)

P_d - diametral pitch (in^{-1})

PV   - pressure times velocity scoring factor (MPa - m/s)

Q_v - AGMA surface quality factor

R   - gear radius (in)

R - reliability

R_C - calculated reliability

R_D - measured reliability

S_{ac} - surface compression strength (psi)

S_{se} - shear endurance strength (MPa)

S_{su} - shear ultimate strength (MPa)

S_{sy} - shear yield strength (MPa)

S_{uc} - tensile strength constant (MPa)

139

ΔS	– step size
T_B	– base temperature (°F)
T_f	– flash temperature (°F)
T_f	– friction torque (N-m)
T_q	– applied torque (lb-in)
V	– shear force (N)
V_{coil}	– spring coil cylinder volume (mm^3)
V_k	– single violated constraint
V_w	– wire volume (mm^3)
V	– pitch line velocity (ft/sec)
V_s	– sliding velocity (m/s) or (ft/sec)
∇v_k	– single violated constraint gradient
w	– weight density (kN/m^3)
W	– gear weight (lbs)
W_t	– spring weight (N)
X	– independent design parameter vector
XD_i	– life test data value (10^3 h)
X_L	– independent design parameter lower value
X_U	– independent design parameter upper value
X_G	– geometric temperature factor
X_M	– thermal-elastic temperature factor
X_Γ	– load sharing temperature factor
Y	– scaled independent design parameter vector
ΔY	– incremental step change in scaled design parameter vector
YD_i	– reliability data point value
α	– gear tooth involute angle (radians)

β — length to diameter ratio

δ — spring deflection (mm)

δ_1 — involute interference angle (radians)

θ — spring wire rotation angle (radians)

μ — coefficient of friction

μ_m — surface roughness (RMS)

ν — Poisson's ratio

ρ — radius of curvature (in)

σ_b — bending stress (psi)

σ_D — dispersion of data from fitted curve

σ_H — Hertzian contact stress (psi)

σ_{Ht} — Hertzian contact stress at the gear tooth tip (psi)

τ — shear stress (MPa)

ϕ — pressure angle (radians)

ω — gear angular velocity (rad/sec)

Ω — shaft speed (RPM)

Subscripts

a — alternating

a1 — pinion addendum

a2 — gear addendum

b1 — pinion base

b2 — gear base

d — dynamic

i — independent design parameter index

j — optimization step index

141

k   - violated constraint index

m   - mean

max   - maximum

min   - minimum

sol   - solid height

1   - pinion

2   - gear

10   - 90 percent reliability

Superscripts

a   - wire strength exponent

b   - Weibull slope

p   - load-life exponent

References

1. Fox, R.L., <u>Optimization Methods for Engineering Design</u>, Addison Wesley, 1971.

2. Vanderplaats, G.N., <u>Numerical Optimization Techniques for Engineering Design: with Applications</u>, McGraw Hill, 1984.

3. Papalambros, P.Y. and Wilde, D.J., <u>Principles of Optimal Design, Modeling and Computation</u>, Cambridge University Press, 1988.

4. Savage, M., Prasanna, M. G. and Coe, H. H., "Maximum Life Spiral Bevel Reduction Design," <u>Gear Technology, The Journal of Gear Manufacturing</u>, Vol. 10. No. 5, September - October 1993, pp. 24-40.

5. Joachim, F. J., "Application of Modern Calculation Methods to Gears in Vehicle Transmissions," <u>Proceedings of The International Conference on Motion And Power Transmission</u>, Hiroshima, Japan, November 1991, pp. 583 - 588.

6. Johnson, R. C., <u>Mechanical Design Synthesis</u>, 2nd Edition, Krieger Publishing Co., Huntington, New York, 1978.

7. Savage, M., Lattime, S. B., Kimmel, J. A. and Coe, H. H., "Optimal Design of Compact Spur Gear Reductions," <u>ASME Journal of Mechanical Design</u>, Vol. 116, No. 3, September 1994, pp. 690-696.

8. Savage, M., Mackulin, B. J., Coe H. H. and Coy, J. J., "Maximum Life Spur Gear Design," <u>AIAA Journal of Propulsion and Power</u>, Vol. 8. No. 6, November - December 1992, pp. 1273-1281.

9. Mott, R. L., <u>Machine Elements in Machine Design</u>, 2nd Edition, Macmillan, New York, 1992.

10. Juvinall, R. C. & Marshek, K. M., <u>Fundamentals of Machine Component Design</u>, Wiley, New York, 2nd Ed., 1991.

11. Green, R., <u>Machinery Handbook</u>, 24th Edition, Industrial Press, New York, 1992.

12. <u>Microsoft Fortran, Version 5.1</u>, Microsoft Corp., Redmond, Washington, 1991.

13. <u>Lahey F77L Fortran</u>, Lahey Computer Systems, Inc., Incline Village, Nevada, 1992.

14. Errichello, R., "A Rational Procedure for Designing Minimum-Weight Gears," <u>Proceedings of the 1989 International Power Transmission and Gearing Conference</u>, ASME, Chicago, Illinois, April 25-28, 1989, pp. 111-114.

15. Shigley, J. E., and Mischke, C. R., <u>Mechanical Engineering Design</u>, 5th Edition, Mcgraw Hill, New York, 1989.

16. Savage, M., Coy, J.J., and Townsend, D.P., "Optimal Tooth Numbers for Compact Standard Spur Gear Sets," <u>ASME Journal of Mechanical Design</u>, Vol. 104, No. 4, Oct. 1982, pp. 749-758.

17. AGMA STANDARD, "Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth," ANSI/AGMA 2001-B88, Alexandria, Va., Sept. 1988.

18. Townsend, D. P., Editor, <u>Dudley's Gear Handbook</u>, 2nd Edition, McGraw Hill, New York, 1991.

19. Astridge, D. and Savage, M., <u>Rotorcraft Drivetrain Life Safety and Reliability,</u> AGARD REPORT No. 775, NATO, Loughton, Essex, June 1990.

20. Buckingham, E. K., <u>Analytical Mechanics Of Gears</u>, McGraw Hill, New York, 1949.

21. Coy, J. J., Townsend, D. P., and Zaretsky, E. V., "Experimental and Analytical Load-Life Relations For AISI 9310 Steel Spur Gears", <u>ASME Journal of Mechanical Design</u>, Vol 100, No. 1, January 1978, pp. 54 - 60.

22. Schenck, H., <u>Theories of Engineering Experimentation</u>, 3rd Edition, McGraw Hill, New York, 1979.

23. Lipson, C. and Sheth, N. J., <u>Statistical Design and Analysis of Engineering Experiments</u>, McGraw Hill, New York, 1973.

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1995	Final Contractor Report

4. TITLE AND SUBTITLE

SEEK—A Fortran Optimization Program Using a Feasible Directions Gradient Search

6. AUTHOR(S)

M. Savage

5. FUNDING NUMBERS

WU–505–62–36
G–NAG3–1047
1L162211A47A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The University of Akron
Akron, Ohio 44325

8. PERFORMING ORGANIZATION REPORT NUMBER

E–9338

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Vehicle Propulsion Directorate
U.S. Army Research Laboratory
Cleveland, Ohio 44135–3191
and
NASA Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–4641
ARL–TR–645

11. SUPPLEMENTARY NOTES

Project Manager, Harold H. Coe, Propulsion Systems Division, NASA Lewis Research Center, organization code 2730, (216) 433–3971.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 37

This publication is available from the NASA Center for Aerospace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report describes the use of computer program 'SEEK' which works in conjunction with two user-written subroutines and an input data file to perform an optimization procedure on a user's problem. The optimization method uses a modified feasible directions gradient technique. SEEK is written in ANSI standard Fortran 77, has an object size of about 46 K bytes and can be used on a personal computer running DOS. This report describes the use of the program and discusses the optimizing method. The program use is illustrated with four example problems: a bushing design, a helical coil spring design, a gear mesh design and a two-parameter Weibull life-reliability curve fit.

14. SUBJECT TERMS

Optimization; Gears; Computer programs; Fatigue life; Design

15. NUMBER OF PAGES

149

16. PRICE CODE

A07

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		